```
                        .TITLE  "VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM"
                ;
                PAL     =       0               ;0 => NTSC, 1=> PAL VERSION
                PRNT    =       0               ;0 => ROM, 1=> DEBUG VERSION WHICH PRINTS ON TI TERMINAL
                STACK   =       0               ;0 => NORMAL ALPHA-BETA "STACK", 1 => EXTRA LARGE STACK
                ROMSTR  =       $F000           ;ROM START ADDR
                ;
                ;       DK1:SCHKG.SRC
                ;       COPYRIGHT ATARI, 1980
                ;       STARTED 10/1/79
                ;
                ;       THE ALPHA-BETA PRUNING ALGORITHM USED HEREIN IS BASED ON THE LISP
                ;       PROGRAM ON PAGE 321 OF "ARTIFICIAL INTELLIGENCE" BY PATRICK HENRY
                ;       WINSTON, 1977, ADDISON-WESLEY PUBLISHING COMPANY, INC. PHILIPPINES.
                ;       ALSO ON "SOME STUDIES IN MACHINE LEARNING USING THE GAME OF CHECKERS",
                ;       BY A. L. SAMUEL, PARTS I AND II, IN THE IBM JOURNAL OF RESEARCH AND DEV
                ;       JULY 1959 AND NOVEMBER 1967.  PART I IS ALSO IN "COMPUTERS AND THOUGH
                ;       BY EDWARD A. FEIGENBAUM AND JULIAN FELDMAN, PP.71-105, 1963, MCGRAW
                ;
                ;       STELLA (TIA) REGISTER ADDRESSES
                ;
0000            VSYNC   =       $00             ;BIT     1  VERTICAL SYNC SET-CLR
0001            VBLANK  =       $01             ;BIT  76 1  VERTICAL BLANK SET-CLR,
0002            WSYNC   =       $02             ;STROBE     WAIT FOR HORIZ BLANK
0003            RSYNC   =       $03             ;STROBE     RESET HORIZ SYNC CO
0004            NUSIZ0  =       $04             ;BITS    54 210 NUMBER-SIZE PLAYER
0005            NUSIZ1  =       $05             ;BITS    54 210 NUMBER-SIZE PLAY
0006            COLUP0  =       $06             ;BITS 7654321  COLOR(4)-LUM(3
                ;                               ;BIT7=1 => POT CAPACI
                INPT0   =       $38             ;POT 0.
                INPT1   =       $39             ;POT 1
                INPT2   =       $3A             ;POT 2
                INPT3   =       $3B             ;POT 3
                INPT4   =       $3C             ;JOYSTICK 0 BUTTON.
                INPT5   =       $3D             ;JOYSTICK 1 BUTTON
                ;
                ;       PIA AND TIMER (6532) LOCATIONS
                ;
                SWCHA   =       $280            ;P0,P1 JOYS
                SWACNT  =       $281            ;SWCHA I/O
                SWCHB   =       $282            ;CONSOL
                INTIM   =       $284            ;T
```

```
                        .TITLE  "VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM"
                ;
0000            PAL     =       0               ;0 => NTSC,  1=> PAL VERSION
0000            PRNT    =       0               ;0 => ROM,   1=> DEBUG VERSION WHICH PRINTS ON TI TERMINAL
0000            STACK   =       0               ;0 => NORMAL ALPHA-BETA "STACK",  1 => EXTRA LARGE STACK
F000            ROMSTR  =       $F000           ;ROM START ADDR
                ;
                ;       DK1:SCHKG.SRC
                ;       COPYRIGHT ATARI, 1980
                ;       STARTED 10/1/79
                ;
                ;       THE ALPHA-BETA PRUNING ALGORITHM USED HEREIN IS BASED ON THE LISP
                ;       PROGRAM ON PAGE 321 OF "ARTIFICIAL INTELLIGENCE" BY PATRICK HENRY
                ;       WINSTON, 1977, ADDISON-WESLEY PUBLISHING COMPANY, INC. PHILIPPINES.
                ;       ALSO ON "SOME STUDIES IN MACHINE LEARNING USING THE GAME OF CHECKERS",
                ;       BY A. L. SAMUEL, PARTS I AND II, IN THE IBM JOURNAL OF RESEARCH AND DEVELOPMENT,
                ;       JULY 1959 AND NOVEMBER 1967.  PART I IS ALSO IN "COMPUTERS AND THOUGHT",
                ;       BY EDWARD A. FEIGENBAUM AND JULIAN FELDMAN, PP. 71-105, 1963, MCGRAW-HILL.
                ;
                ;
                ;       STELLA (TIA) REGISTER ADDRESSES
                ;
0000            VSYNC   =       $00     ;BIT         1  VERTICAL SYNC SET-CLR     1=>START
0001            VBLANK  =       $01     ;BIT  76     1  VERTICAL BLANK SET-CLR, PORT CONTROL (INPT0-INPT5)
0002            WSYNC   =       $02     ;STROBE         WAIT FOR HORIZ BLANK
0003            RSYNC   =       $03     ;STROBE         RESET HORIZ SYNC COUNTER
0004            NUSIZ0  =       $04     ;BITS   54 210  NUMBER-SIZE PLAYER/MISSILE 0
0005            NUSIZ1  =       $05     ;BITS   54 210  NUMBER-SIZE PLAYER/MISSILE 1
0006            COLUP0  =       $06     ;BITS 7654321   COLOR(4)-LUM(3) PLAYER 0
0007            COLUP1  =       $07     ;BITS 7654321   COLOR(4)-LUM(3) PLAYER 1
0008            COLUPF  =       $08     ;BITS 7654321   COLOR(4)-LUM(3) PLAYFIELD
0009            COLUBK  =       $09     ;BITS 7654321   COLOR(4)-LUM(3) BACKGROUND
000A            CTRLPF  =       $0A     ;BITS   54 210  PLAYFIELD CONTROL & BALL
000B            REFP0   =       $0B     ;BIT      3     REFLECT PLAYER 0.  1=>REFLECT
000C            REFP1   =       $0C     ;BIT      3     REFLECT PLAYER 1
000D            PF0     =       $0D     ;BITS 7654      PLAYFIELD REG BYTE 0 (REVERSED)
000E            PF1     =       $0E     ;BITS ALL       PLAYFIELD REG BYTE 1
000F            PF2     =       $0F     ;BITS ALL       PLAYFIELD REG BYTE 2 (REVERSED)
0010            RESP0   =       $10     ;STROBE         RESET PLAYER 0
0011            RESP1   =       $11     ;STROBE         RESET PLAYER 1
0012            RESM0   =       $12     ;STROBE         RESET MISSILE 0
0013            RESM1   =       $13     ;STROBE         RESET MISSILE 1
0014            RESBL   =       $14     ;STROBE         RESET BALL
0015            AUDC0   =       $15     ;BITS     3210  AUDIO CONTROL 0
0016            AUDC1   =       $16     ;BITS     3210  AUDIO CONTROL 1
0017            AUDF0   =       $17     ;BITS     3210  AUDIO FREQUENCY 0
0018            AUDF1   =       $18     ;BITS     3210  AUDIO FREQUENCY 1
0019            AUDV0   =       $19     ;BITS     3210  AUDIO VOLUME 0
001A            AUDV1   =       $1A     ;BITS     3210  AUDIO VOLUME 1
```

```
001B                    GRP0    =       $1B     ;BITS ALL       GRAPHICS FOR PLAYER 0
001C                    GRP1    =       $1C     ;BITS ALL       GRAPHICS FOR PLAYER 1
001D                    ENAM0   =       $1D     ;BIT         1  ENABLE MISSILE 0.  1=> ENABLE
001E                    ENAM1   =       $1E     ;BIT         1  ENABLE MISSILE 1
001F                    ENABL   =       $1F     ;BIT         1  ENABLE BALL
0020                    HMP0    =       $20     ;BITS 7654      HORIZ MOTION PLAYER 0.  0-7 => LEFT
0021                    HMP1    =       $21     ;BITS 7654      HORIZ MOTION PLAYER 1.  F-8 => RIGHT
0022                    HMM0    =       $22     ;BITS 7654      HORIZ MOTION MISSILE 0
0023                    HMM1    =       $23     ;BITS 7654      HORIZ MOTION MISSILE 1
0024                    HMBL    =       $24     ;BITS 7654      HORIZ MOTION BALL
0025                    VDELP0  =       $25     ;BIT         0  VERTICAL DELAY PLAYER 0.  1=> VDEL ON
0026                    VDELP1  =       $26     ;BIT         0  VERTICAL DELAY PLAYER 1
0027                    VDELBL  =       $27     ;BIT         0  VERTICAL DELAY BALL
0028                    RESMP0  =       $28     ;BIT         1  RESET MISSILE TO PLAYER 0.  1=> MISSILE RESET TO PLAYER
0029                    RESMP1  =       $29     ;BIT         1  RESET MISSILE TO PLAYER 1
002A                    HMOVE   =       $2A     ;STROBE         ACT ON HORIZ MOTION
002B                    HMCLR   =       $2B     ;STROBE         CLEAR ALL HM REGISTERS
002C                    CXCLR   =       $2C     ;STROBE         CLEAR COLLISION LATCHES
                        ;
                        ;       READ ADDRESSES - BITS 7 & 6 ONLY
                        ;
0030                    CXM0P   =       $30     ;M0*P1  M0*P0
0031                    CXM1P   =       $31     ;M1*P0  M1*P1
0032                    CXP0FB  =       $32     ;P0*PF  P0*BL
0033                    CXP1FB  =       $33     ;P1*PF  P1*BL
0034                    CXM0FB  =       $34     ;M0*PF  M0*BL
0035                    CXM1FB  =       $35     ;M1*PF  M1*BL
0036                    CXBLPF  =       $36     ;BL*PF
0037                    CXPPMM  =       $37     ;P0*P1  M0*M1
0038                    INPT0   =       $38     ;POT 0.  BIT7=1 => POT CAPACITOR IS COMPLETELY CHARGED?
0039                    INPT1   =       $39     ;POT 1
003A                    INPT2   =       $3A     ;POT 2
003B                    INPT3   =       $3B     ;POT 3
003C                    INPT4   =       $3C     ;JOYSTICK 0 BUTTON.  B7=0 => BUTTON PRESSED
003D                    INPT5   =       $3D     ;JOYSTICK 1 BUTTON
                        ;
                        ;       PIA AND TIMER (6532) LOCATIONS
                        ;
0280                    SWCHA   =       $280    ;P0,P1 JOYSTICKS (RLDU,RLDU), POT TRIGGERS, KEYBOARD, DRIVING CONTROLLER
0281                    CTLSWA  =       $281    ;SWCHA I/O CONTROL      1=OUTPUT
0282                    SWCHB   =       $282    ;CONSOLE SWITCHES (READ ONLY) P1B,P0B,X,X,B/W,X,SELECT,RESET IF 0
0283                    CTLSWB  =       $283    ;UNUSED
0284                    INTIM   =       $284    ;INTERVAL TIMER INPUT   0=>TIMER UP
0294                    TIM1T   =       $294    ;TIMER OUTPUT:  1 MACHINE CYCLE/TICK (.838 MICROSEC)
0295                    TIM8T   =       $295    ;8 MACHINE CYCLES/TICK
0296                    TIM64T  =       $296    ;64 MC'S/TICK
0297                    T1024T  =       $297    ;1024 MACHINE CYCLES/TICK (858.2 MICROSECONDS)
                        ;
                        ;       KIM SUBROUTINES
                        ;
1E2F                    CRLF    =       $1E2F   ;PRINT CRLF ON TI
1E3B                    PRTBYT  =       $1E3B   ;PRINT ACCUMULATOR IN HEX ASCII ON TI
1E5A                    GETCH   =       $1E5A   ;READ ASCII CHAR FROM TI INTO A
1E9E                    OUTSP   =       $1E9E   ;PRINT SPACE ON TI
1EA0                    OUTCH   =       $1EA0   ;PRINT ACCUMULATOR AS ASCII CHAR ON TI
1F9D                    GETBYT  =       $1F9D   ;READ TWO HEX ASCII CHARS FROM TI INTO A
```

```
                        ;                       CHECKERS EQUATES
000B                    MAXMAX  =       11              ;MAX STACK DEPTH FOR ALPHA-BETA
                                .IF     STACK
                        MAXMAX  =       22
                                .ENDIF

0009                    JMPLEN  =       9               ;LENGTH OF JMPTBL (LIST OF JUMPED PIECES)

0010                    LINCHR  =       16              ;# OF LINES/CHAR
0000                    EMPTY   =       0               ;EMPTY SQUARE INDEX INTO CHARACTER TABLE
0010                    CHECKER =       1*LINCHR
0020                    KING    =       2*LINCHR
0030                    CURS    =       3*LINCHR

00AA                    BLANK   =       10*16+10        ;TWO BLANK CHARS
00BB                    JP      =       11*16+11        ;"JP"

0000                    CLP0    =       0               ;PLAYER 0 (HUMAN, LEFT JOYSTICK)
0080                    CLP1    =       $80             ;PLAYER 1 (COMPUTER, RIGHT PLAYER)

0010                    HUMCHK  =       CLP0+CHECKER    ;HUMAN CHECKER (NOT KING)
0020                    HUMKNG  =       CLP0+KING       ;HUMAN KING
0030                    HUMCRS  =       CLP0+CURS       ;HUMAN CURSOR
0090                    CMPCHK  =       CLP1+CHECKER    ;COMPUTER CHECKER
00A0                    CMPKNG  =       CLP1+KING       ;COMPUTER KING
00B0                    CMPCRS  =       CLP1+CURS       ;COMPUTER CURSOR

0005                    HMIN    =       5               ;MIN. GAMNO FOR HARD BOARD COLORS

0036                    DARK    =       $36             ;DARK CHECKER COLOR/LUM
000C                    LIGHT   =       $0C             ;LIGHT CHECKER COLOR/LUM
0000                    EDARK   =       0               ;EASY DARK SQUARE COLOR.  0 IS ACTUALLY USED INSTEAD OF EDARK
0034                    ELIGHT  =       $34             ;EASY LIGHT SQUARE COLOR (RED)
0036                    ELIAT   =       $36             ;EASY LIGHT IF ATTRACT
00B0                    HDARK   =       $B0             ;HARD DARK SQUARE COLOR (GREEN)
0026                    HLIGHT  =       $26             ;HARD LIGHT SQUARE COLOR (BUFF OR CREAM OR WHITE)
0026                    HLIAT   =       $26             ;HARD LIGHT SQUARE IF ATTRACT
000C                    LTCHR   =       $0C             ;LIGHT CHAR COLOR
0092                    DRKCHR  =       $92             ;DARK CHARACTER COLOR
002D                    GSCON   =       45              ;GAME SELECT TIMER INITIAL VALUE
0025                    OVRCON  =       $25             ;OVERSCAN TIMER INITIAL VALUE
002C                    VBLCON  =       44              ;VBLANK TIMER INITIAL VALUE
001E                    VBTIM   =       30              ;1/2 SEC. VBLANK AFTER CALCULATION -- CHANGE FOR PAL
002D                    FRZTIM  =       45              ;FREEZE TIMER FOR END OF HUMAN MOVE

                        ;
                        ;       PAL VERSIONS OF THE ABOVE
                        ;
                                .IF     PAL
                        DARK    =       $66             ;DARK CHECKER COLOR/LUM
                        LIGHT   =       $0C             ;LIGHT CHECKER COLOR/LUM
                        EDARK   =       0               ;EASY DARK SQUARE COLOR
                        ELIGHT  =       $64             ;EASY LIGHT SQUARE COLOR (RED)
                        ELIAT   =       $66             ;EASY LIGHT IF ATTRACT
                        HDARK   =       $50             ;HARD DARK SQUARE COLOR (GREEN)
                        HLIGHT  =       $46             ;HARD LIGHT SQUARE COLOR (BUFF OR CREAM OR WHITE)
                        HLIAT   =       $46             ;HARD LIGHT SQUARE IF ATTRACT
                        LTCHR   =       $0C             ;LIGHT CHAR COLOR
                        DRKCHR  =       $B2             ;DARK CHARACTER COLOR
                        GSCON   =       38              ;GAME SELECT TIMER INITIAL VALUE
```

```
                        OVRCON  =       $2B             ;OVERSCAN TIMER INITIAL VALUE
                        VBLCON  =       $36             ;VBLANK TIMER INITIAL VALUE
                        VBTIM   =       $32             ;1/2 SEC. VBLANK AFTER CALCULATION -- CHANGE FOR PAL
                        FRZTIM  =       38              ;FREEZE TIMER FOR END OF HUMAN MOVE
                                .ENDIF


FFFB                    RB      =       -5              ;SQUARE # OFFSET FOR RIGHT BACKWARD (DOWN) MOVEMENT
FFFC                    LB      =       -4              ;                       LEFT  BACKWARD
0004                    RF      =       4               ;                       RIGHT FORWARD
0005                    LF      =       5               ;                       LEFT  FORWARD

0010                    PLAY2   =       $10             ;TWO-PLAYER GAME # IN DECIMAL

0036                    DROP    =       $36             ;PIECE DROPPING SOUND FOR AUDV0
0030                    PICKUP  =       $30             ;PIECE PICKUP SOUND FOR AUDV0
0004                    PURE2   =       4               ;PURE TONE/4 FOR AUDC0
000F                    ERRSND  =       15              ;ERROR SOUND FOR AUDC0

0008                    MLT     =       $08             ;MULTIPLE JUMP MASK
0004                    MST     =       $04             ;MUST JUMP MASK

0004                    CORN    =       4               ;DOUBLE CORNER CREDIT

0008                    ENDPC   =       4*2             ;MIN NUMBER OF HUMAN PIECES FOR NON-END GAME

0020                    ADDTRM  =       $20             ;TERM TO ADD TO MAGNITUDE OF SCORE IF STACK IS FULL IN TERMNODE
```

```
                        ;                       VARIABLES (RAM PAGE ZERO)
                                *=$80
                        ;                       THE FOLLOWING RAM LOCS ARE CLEARED DURING RESET:
                        ;A-B MEANS USED BY ALPHA-BETA PRUNING ROUTINE
                        ;SAVE MEANS ALPHA-BETA PRUNING ROUTINE DOES NOT ALTER
0080                    ZROBGN
0080                    BOARD
                                *=*+35          ;CHECKER BOARD (1 BYTE / SQUARE) SQUARES 8,17,26 ARE NOT USED.  A-B
00A3                    PIECE
                                *=*+1           ;CURRENT PIECE BEING MOVED.  A-B

00A4                    JMPI
                                *=*+1           ;INDEX INTO JMPLST --  <0 => NO JUMPS.  A-B
00A5                    MSTJMP
                                *=*+1           ;0 => ACTIVE PLAYER CAN'T JUMP   4 => ACTIVE PLAYER MUST JUMP.   A-B
00A6                    MLTJMP
                                *=*+1           ;8 => CONTINUING MULTIPLE JUMP.  A-B & MAIN
00A7                    MSTFLG
                                *=*+1           ;MSTJMP SAVED WHILE NEW MSTJMP IS COMPUTED.  A-B
00A8                    MLTFLG
                                *=*+1           ;MLTJMP BACKUP.  A-B

00A9                    HPIECE
                                *=*+1           ;HUMAN PIECE COUNT.   KING=3, CHECKER=2.  A-B
00AA                    CPIECE
                                *=*+1           ;COMPUTER PIECE COUNT.  A-B

00AB                    CURSOR
                                *=*+1           ;SQUARE # FOR BLINKING CURSOR.  SAVE
00AC                    FROMB
                                *=*+1           ;BINARY "FROM" SQUARE (INTERNAL NUMBERING).  A-B
00AD                    TOSQRB
                                *=*+1           ;BINARY "TO" SQUARE.   A-B

00AE                    HKING
                                *=*+1           ;HUMAN KING COUNT
00AF                    CKING
                                *=*+1           ;COMPUTER KING COUNT (1 FOR EACH KING)

00B0                    ZROEND
                        ;
                        ;                       THE FOLLOWING RAM LOCS ARE NOT CLEARED DURING RESET:
                        ;
                        ; GAMNO = 1 TO GMAX2 IN DECIMAL
                        ;       1-9             REGULAR CHECKERS
                        ;       10              2-PLAYER (EITHER)
                        ;       11-19           GIVE-AWAY (LOSING)
00B0                    GAMNO
                                *=*+1           ;GAME NUMBER
00B1                    COLHUM
                                *=*+1           ;HUMAN CHECKER COLOR (NO ATTRACT ADDED IN).  SAVE
00B2                    COLCMP
                                *=*+1           ;COMPUTER CHECKER COLOR (NO ATTRACT ADDED IN).  SAVE?
                        ;
00B3                    TO
                                *=*+1           ;TEMP (USED IN  KERNEL, ETC.).  A-B & MAIN
00B4                    T1
                                *=*+1           ;TEMP (USED IN KERNEL, ETC.).  A-B & MAIN
00B4                    XSAVE   =       T1      ;X REG SAVED FOR JMPCHK SUBROUTINE.  A-B & MAIN
00B5                    T2
```

```
                        *=*+1           ;A-B & MAIN
00B5        YSAVE       =       T2      ;. A-B & MAIN
00B5        NUM         =       T2      ;2 BYTE NUMERATOR FOR STATIC EVALUATION DIVIDE (USED FOR RATIO OF PIECE COUNTS). A-B
00B6        T3
                        *=*+1           ;A-B & MAIN
00B6        ANYMVE      =       T3      ;>0 => ACTIVE PLAYER CAN MOVE.  A-B & MAIN -- USED ONLY BY JMPCHK
00B7        T6
                        *=*+1           ;ANOTHER TEMP
00B8        DEN
                        *=*+1           ;DENOMINATOR.  A-B
00B9        T7
                        *=*+1           ;TEMP
00BA        T8
                        *=*+1           ;YET ANOTHER TEMP -- USED TO SAVE BK COLOR

00BB        ACTIVE
                        *=*+1           ;ACTIVE OR CURRENT PLAYER (HUMAN=0, COMPUTER=$80).  A-B & MAIN
00BC        DEPTH
                        *=*+1           ;CURRENT DEPTH OF TREE SEARCH.  A-B & SET UP BY MAIN

00BD        JMPSQR
                        *=*+1           ;SQUARE # FOR PIECE BEING JUMPED.  A-B
00BE        INDEX
                        *=*+1           ;CURRENT OFFSET INDEX (0-3).  A-B
00BF        FRSMOV
                        *=*+1           ;$80 => FIRST MOVE (MAY BE MULTIPLE JUMP).  A-B
                ;       B7=1 => FIRST MOVE, B6=1 => FIRST MOVE AT THIS LEVEL, BETTER MOVE FOUND
00C0        XMOVE
                        *=*+1           ;"FROM" MOVE SQUARE FROM MOVCHK.  A-B & MAIN
00C1        TIMOUT
                        *=*+1           ;TIMER

00C2        OLDPBQ
                        *=*+1           ;SWCHB FOR THIS FRAME.
00C3        FRAME
                        *=*+1           ;FRAME COUNTER -- COUNTS UP.
00C4        PRNCNT
                        *=*+1           ;B7=1 => PRUNE (FOR MULTIPLE JUMPS).  B6-B0 = RANDOM MOVE COUNTER

00C5        JMPLST
                        *=*+JMPLEN      ;SQUARES FOR PIECES JUMPED BY COMPUTER, <0 => KING.  A-B & MAIN
00CE        JMPEND
                ;
                ;       THE FOLLOWING ALPHA-BETA "STACK" OVERLAP THE MAIN PROGRAM VARS WHICH
                ;       ARE LISTED LAST.  THEY ALSO OVERLAP JMPLST WITH THE EXCEPTION OF THE FIRST 3 BYTES.
                ;
                        *=JMPLST+3
                        .IF     STACK
                AX=*
                        *=$180
                        .ENDIF          ;ALLOW FOR BIGGER STACK IF DEBUG
00C8        FROMT
                        *=*+MAXMAX      ;FROM(I-1) = FROM SQUARE FOR DEPTH I.  A-B
                ;       B7=1 => FIRST MOVE, B6=1 => FIRST MOVE AT THIS LEVEL, BETTER MOVE FOUND
00D3        OTHER
                        *=*+MAXMAX      ;OTHER(I-1) = OTHER FOR DEPTH I.  A-B
                ; B7-6 = CAPTURED PIECE, B5-4 = MOVED PIECE, B3 = MLTJMP, B2 = MSTJMP, B1-0 = OFFSET INDEX
00DE        ALPHAL
                        *=*+MAXMAX+1+1  ;ALPHAL(I) = ALPHA(DEPTH) LSB.  A-B
00EB        ALPHAH
```

```
                        *=*+MAXMAX+1+1   ;ALPHAH(I) = ALPHA(DEPTH)/256.  A-B
00F8                    ALPEND

                        .IF      STACK
                        *=AX
                        .ENDIF

                    ;
                    ;                           THE FOLLOWING VARS OVERLAP THE ALPHA-BETA "STACK" AND T4-SCRRPF:
                    ;
                        *=JMPLST+JMPLEN
00CE                    COLP0
                        *=*+1               ;COLOR OF P0 (ATTRACT ADDED IN)
00CF                    COLP1
                        *=*+1               ;COLOR OF P1

00D0                    COL0
                        *=*+2               ;COLOR OF 1ST PIECE IN ROW (FOR KERNEL ONLY)
00D2                    COL1
                        *=*+2
00D4                    COL2
                        *=*+2
00D6                    COL3
                        *=*+2

00D8                    PNTR0
                        *=*+2               ;POINTERS TO SQUARE GRAPHICS
00DA                    PNTR1
                        *=*+2
00DC                    PNTR2
                        *=*+2
00DE                    PNTR3
                        *=*+2

00E0                    SQUARE
                        *=*+1               ;SQUARE # FOR USE IN KERNEL
00E1                    SQREND
                    ;
                    ;       THE FOLLOWING PARTIALLY OVERLAP BOTH THE ALPHA-BETA "STACK" AND
                    ;       COLP0-SQUARE.
                    ;
                        *=JMPLST+JMPLEN
00CE                    T4
                        *=*+1
00CF                    T5
                        *=*+1

00D0                    SCRLP0
                        *=*+5               ;P0 "SCORE" GRAPHICS
00D5                    SCRRP1
                        *=*+5               ;P1
00DA                    SCRLPF
                        *=*+5               ;LEFT PF1 GRAPHICS
00DF                    SCRRPF
                        *=*+5               ;RIGHT PF1 GRAPHICS

00E4                    MOVVAL
                        *=*+1               ;VALUE IN MOVE SQUARE.  SET UP AT END OF A-B.  DON'T SAVE
00E5                    MOVFLG
                        *=*+1               ;1 => DISPLAY COMPUTER'S MOVE.  DON'T SAVE
```

```
00E6                    WINSAV
                        *=*+1           ;WINNING PLAYER, 0=> NO WIN.  DON'T SAVE
00E7                    FROMTO
                        *=*+1           ;0 => FROM, 1=> TO.  DON'T SAVE

00E8                    CURSC
                        *=*+1           ;COMPUTER CURSOR.  SET UP AT END OF A-B.  DON'T SAVE
00E9                    MOVE
                        *=*+1           ;CURRENT MOVE.  SET UP AT END OF A-B.  DON'T SAVE
00EA                    OLDINP
                        *=*+1           ;JOYSTICK BUTTON VALUE FOR CURRENT PLAYER FOR LAST FRAME.  DON'T SAVE
00EB                    GSTIM
                        *=*+1           ;GAME SELECT TIMER -- COUNTS DOWN TO 0.  DON'T SAVE

00EC                    OLDPB6
                        *=*+1           ;OLD SWCHB VALUE, 0=>NORMAL, <>0=>SETUP (BIT 6 ONLY).  DON'T SAVE
00ED                    ILEGAL
                        *=*+1           ;>0 => MAKE ILLEGAL MOVE SOUND.  DON'T SAVE
00EE                    ATIM
                        *=*+1           ;ATTRACT MODE TIMER -- COUNTS UP.  DON'T SAVE
00EF                    GFLG2
                        *=*+1           ;0=>ATTRACT  <> 0 => NOT ATTRACT.  DON'T SAVE
00F0                    BLNKTM
                        *=*+1           ;FRAME COUNTER FOR VBLANK AFTER CALCULATION(DOWN TO 0).  DON'T SAVE
00F1                    SNDTIM
                        *=*+1           ;TIMER FOR SOUNDS (COUNTS DOWN TO 0).  DON'T SAVE
00F2                    FREEZE
                        *=*+1           ;TIMER FOR FREEZE AFTER HUMAN MOVE (0=END OF FREEZE)
00F3                    COLSQ
                        *=*+1           ;SQUARE COLOR FOR KERNEL


00F4                    FROM
                        *=*+1           ;FROM SQUARE FOR "SCORE" .  DON'T SAVE
00F5                    TOSQR
                        *=*+1           ;TO SQUARE FOR "SCORE".  DON'T SAVE
00F6                    SCP0
                        *=*+1           ;GAME # FOR "SCORE".  SAVE
00F7                    SCP1
                        *=*+1           ;# OF PLAYERS FOR "SCORE".  DON'T SAVE

                        ;               F8-FF ARE USED FOR STACK (4 LEVELS DEEP FOR NOW)
```

```
                        ;                                       SCORE KERNEL (TOP LINE OF CHARS)
                                *=ROMSTR
                                .IF     PRNT
                                *=$E000                         ;RELOCATE IF DEBUG TO MAKE ROOM FOR MORE CODE
                                JMP     PSTART
                                .ENDIF

F000                    JMPBAK
F000 A2 04                      LDX     #5-1                    ;5 BYTES PER CHARACTER
F002                    SCOR2
                        ;                                       MACHINE CYCLE COUNTS ARE GIVEN
F002 A0 02                      LDY     #2                      ;50 49 REPEAT EACH BYTE TWICE
F004                    SCOR3
F004 85 02                      STA     WSYNC                   ;47 53 46 52
F006 B5 DA                      LDA     SCRLPF,X                ;4
F008 85 0E                      STA     PF1                     ;7  GAME # (AT LEFT)
F00A B5 D0                      LDA     SCRLP0,X                ;11
F00C 85 1B                      STA     GRP0                    ;14 "FROM" SQUARE
F00E B5 D5                      LDA     SCRRP1,X                ;18
F010 85 1C                      STA     GRP1                    ;21 "TO" SQUARE

F012 A9 00                      LDA     #0                      ;23 MISSILE 0 FOR DASH BETWEEN "FROM" AND "TO"
F014 E0 02                      CPX     #2                      ;25  X=0 ON LAST TIME THROUGH SO CARRY IS CLEAR
F016 D0 01                      BNE     SCOR4                   ;27 28
F018 8A                         TXA                             ;29 X=2
F019                    SCOR4
F019 85 1D                      STA     ENAM0                   ;32 31

F01B B5 DF                      LDA     SCRRPF,X                ;36 35
F01D 88                         DEY                             ;38 37
F01E 85 0E                      STA     PF1                     ;41 40 # OF PLAYERS (AT RIGHT)
F020 D0 E2                      BNE     SCOR3                   ;44 43 43 42

F022 CA                         DEX                             ;45 44
F023 10 DD                      BPL     SCOR2                   ;48 47 46  X=$FF  Y=0
```

```
                        ;       CHECKERBOARD KERNEL
                        ;
                        ;                               X=$FF    Y=0
                        ;                               CARRY IS CLEAR => LAST SQUARES AT LEFT,
                        ;                               NEXT SQUARES AT RIGHT.
F025 A5 F3                      LDA     COLSQ           ;49
F027 85 02                      STA     WSYNC           ;52
F029 85 07                      STA     COLUP1          ;3   SET UP P1 COLOR FOR TOP RIGHT OF TOP RIGHT SQUARE
F02B A9 31                      LDA     #$31            ;5  8 CLOCK BALL, REFLECT PLAYFIELD
F02D 85 0A                      STA     CTRLPF          ;8
F02F 84 0E                      STY     PF1             ;11
F031 84 1B                      STY     GRP0            ;14
F033 84 1C                      STY     GRP1            ;17
                        ;                               SET HORIZONTAL POSITION OF PLAYERS, MISSILE AND BALL
F035 A9 E0                      LDA     #$E0            ;19  2 TO RIGHT
F037 85 24                      STA     HMBL            ;22
F039 8A                         TXA                     ;24  INIT MSB'S OF CHAR POINTERS FOR KERNEL  ($FF)
F03A 85 23                      STA     HMM1            ;27  $FX -- 1 TO RIGHT
F03C 85 D9                      STA     PNTR0+1         ;30
F03E 85 11                      STA     RESP1           ;33    POSITION PLAYER 1 (P1)  (MUST OCCUR AT THIS TIME)
F040 A2 06                      LDX     #6              ;35    NOTE THAT RESET TAKES 4-6 CLOCKS
F042 86 04                      STX     NUSIZ0          ;38  3 COPIES OF PLAYER, MEDIUM SEPARATION
F044 84 21                      STY     HMP1            ;41  Y=0
F046 85 10                      STA     RESP0           ;44    POSITION P0 (MUST OCCUR AT THIS TIME)
F048 A2 30                      LDX     #$30            ;46
F04A 86 05                      STX     NUSIZ1          ;49  8 CLOCK MISSILE, 1 COPY OF PLAYER 0
F04C 85 DB                      STA     PNTR1+1         ;52
F04E 85 DD                      STA     PNTR2+1         ;55
F050 85 DF                      STA     PNTR3+1         ;58
F052 A9 10                      LDA     #$10            ;60  1 TO LEFT
F054 85 20                      STA     HMP0            ;63
F056 85 14                      STA     RESBL           ;66    POSITION BALL (MUST OCCUR AT THIS TIME)
F058 85 13                      STA     RESM1           ;69    POSITION MISSILE #1
F05A 85 02                      STA     WSYNC           ;72 0
F05C 85 2A                      STA     HMOVE           ;3 MOVE OBJECTS A FEW CLOCKS FOR EXACT POSITIONING
F05E A5 BA                      LDA     T8              ;CHANGE BK COLOR
F060 85 09                      STA     COLUBK          ;IF B/W
F062 85 02                      STA     WSYNC           ;12
F064 85 2B                      STA     HMCLR           ;3
F066 A9 C0                      LDA     #$C0            ;5
F068 85 21                      STA     HMP1            ;8   4 TO RIGHT
F06A 85 20                      STA     HMP0            ;11

F06C A0 1F                      LDY     #35-4           ;13  START AT TOP LEFT OF BOARD -- BOARD(31) TO BOARD(35)

                        ;
                        ;       DISPLAY NEXT ROW OF CHECKERS
                        ;
F06E                    LOOP1
F06E 84 E0                      STY     SQUARE          ;16 25
F070 A2 06                      LDX     #3*2            ;18 27  SET UP COLORS AND GRAPHICS FOR 4 CHECKERS IN THIS ROW
                        ;
                        ;       SET UP GRAPHICS IN RAM
                        ;
F072                    KRNLP
F072 B9 80 00                   LDA     BOARD,Y         ;4
F075 30 09                      BMI     KRN10           ;6 7
F077 95 D8                      STA     PNTR0,X         ;10  HUMAN PIECE OR EMPTY
F079 A5 CE                      LDA     COLP0           ;13
F07B 4C 86 F0                   JMP     KRN15           ;16
```

```
F07E B0 50          LOOP15  BCS     LOOP20          ;34  FOR LOOP2 DELAY
F080                KRN10
F080 29 7F                  AND     #$7F            ;9  COMPUTER PIECE
F082 95 D8                  STA     PNTR0,X         ;13
F084 A5 CF                  LDA     COLP1           ;16
F086                KRN15
F086 95 D0                  STA     COL0,X          ;20 20
F088 C8                     INY                     ;22
F089 CA                     DEX                     ;24
F08A CA                     DEX                     ;26
F08B 10 E5                  BPL     KRNLP           ;29  4*29-1=115 M.C.'S
                    ;
                    ;       MIN = 18+115+3 = 136 = 76+60
                    ;       MAX = 27+115+3 = 145 = 76+69
                    ;
                            .IF     PAL             ;PUT TWO EXTRA LINES AT BOTTOM OF SQUARE IF PAL
                            JSR     WAIT2           ;DO 2 WSYNC'S.  JSR & RTS EACH TAKE 6 MC'S
                            .ENDIF
F08D 85 02                  STA     WSYNC           ;3
F08F 85 2A                  STA     HMOVE           ;3  MOVE PLAYERS 4 MORE CLOCKS (RIGHT OR LEFT)
                    ;                               SET UP BALL AND PLAYFIELD
F091 B0 09                  BCS     K100            ;5 6
F093 38                     SEC                     ;7  CS => AT RIGHT
F094 A9 C3                  LDA     #$C3            ;9  PLAYFIELD 2
F096 A2 03                  LDX     #3              ;11  PF1
F098 A0 02                  LDY     #2              ;13  ENABLE BALL
F09A B0 06                  BCS     K110            ;16  JMP
F09C                K100
F09C 18                     CLC                     ;8  CC => AT LEFT
F09D A9 3C                  LDA     #$3C            ;10  PLAYFIELD 2
F09F AA                     TAX                     ;12  PF1
F0A0 A0 00                  LDY     #0              ;14  DISABLE BALL
F0A2                K110
F0A2 85 0F                  STA     PF2             ;19 17
F0A4 86 0E                  STX     PF1             ;22
F0A6 84 1E                  STY     ENAM1           ;25
F0A8 84 1F                  STY     ENABL           ;28

F0AA 85 02                  STA     WSYNC           ;31 29  MOVE 4 MORE CLOCKS
F0AC 85 2A                  STA     HMOVE           ;3
                            .IF     PAL             ;PUT TWO EXTRA LINES AT BOTTOM OF SQUARE IF PAL
                            JSR     WAIT3           ;DO 3 WSYNC'S.  JSR & RTS EACH TAKE 6 MC'S
                            .ENDIF
                            .IF     PAL-1
F0AE 85 02                  STA     WSYNC           ;6 0   WAIT 1 MORE LINE
                            .ENDIF
F0B0 A9 40                  LDA     #$40            ;2      4 TO LEFT
F0B2 B0 02                  BCS     K130            ;4 5
F0B4 A9 C0                  LDA     #$C0            ;6      4 TO RIGHT
F0B6                K130
F0B6 85 20                  STA     HMP0            ;9 8
F0B8 85 21                  STA     HMP1            ;12 11

F0BA A0 0F                  LDY     #LINCHR-1       ;14 13
                                                    DISPLAY ONE ROW OF CHECKER BOARD
                    ;
                    ;
                    ;       DISPLAY NEXT TV LINE
```

```
                        ;
F0BC                    LOOP2
F0BC 85 02                      STA     WSYNC           ;16 15 72 76 0  WAIT FOR NEXT LINE
F0BE A5 D0                      LDA     COL0            ;3 STORE COLOR AND GRAPHICS FOR LEFT CHECKER
F0C0 85 07                      STA     COLUP1          ;6
F0C2 B1 D8                      LDA     (PNTR0),Y       ;11
F0C4 85 1C                      STA     GRP1            ;14

F0C6 A5 D2                      LDA     COL1            ;17  STORE COLOR AND GRAPHICS FOR NEXT CHECKER
F0C8 85 06                      STA     COLUP0          ;20
F0CA B1 DA                      LDA     (PNTR1),Y       ;25
F0CC 85 1B                      STA     GRP0            ;28

F0CE B0 AE                      BCS     LOOP15          ;30 31     4 EXTRA MACHINE CYCLES IF CARRY SET (SQUARES SHIFTED TO RIGHT)
F0D0                    LOOP20
F0D0 A5 F3                      LDA     COLSQ           ;33 37  LOAD DARK SQUARE COLOR
F0D2 85 07                      STA     COLUP1          ;36 40

F0D4 A6 D4                      LDX     COL2            ;39 43  STORE COLOR & GRAPHICS FOR NEXT CHECKER
F0D6 B1 DC                      LDA     (PNTR2),Y       ;44 48
F0D8 86 06                      STX     COLUP0          ;47 51  TIMING IS CRITICAL FROM HERE TO NEXT WSYNC
F0DA 85 1B                      STA     GRP0            ;50 54

F0DC A6 D6                      LDX     COL3            ;53 57  STORE COLOR & GRAPHICS FOR RIGHT CHECKER
F0DE B1 DE                      LDA     (PNTR3),Y       ;58 62
F0E0 85 1B                      STA     GRP0            ;61 65
F0E2 86 06                      STX     COLUP0          ;64 68

F0E4 88                         DEY                     ;66 70
F0E5 10 D5                      BPL     LOOP2           ;69 73 68 72    CONTINUE FOR REST OF LINES IN THIS ROW OF SQUARES

                        ;                               MOVE TO RIGHT OR LEFT 16 CLOCKS FOR NEXT ROW
                        ;                               (4 CLOCKS/MOVE * 4 MOVES)    Y=$FF
F0E7 85 02                      STA     WSYNC           ;71 75
F0E9 85 2A                      STA     HMOVE           ;3  EXECUTE MOVE
F0EB C8                         INY                     ;5  Y=0
F0EC 84 1C                      STY     GRP1            ;8
F0EE 84 1B                      STY     GRP0            ;11

F0F0 85 02                      STA     WSYNC           ;14 0
F0F2 85 2A                      STA     HMOVE           ;3  MOVE SOME MORE

F0F4 08                         PHP                     ;6   SAVE CARRY
F0F5 A5 E0                      LDA     SQUARE          ;9
F0F7 E9 04                      SBC     #4              ;11  SUBTRACT 5 IF CARRY IS CLEAR (COMPENSATES FOR BOARD NUMBERING)
F0F9 28                         PLP                     ;15  RESTORE CARRY
F0FA A8                         TAY                     ;17

F0FB 30 28                      BMI     FRAMLP          ;19
F0FD 4C 6E F0                   JMP     LOOP1           ;22  CONTINUE
                        ;                               END OF KERNELS
```

```
                        ;       INITIALIZATION

F100 78             PSTART  SEI                     ;DISABLE INTERRUPTS
F101 D8                     CLD                     ;BINARY ARITHMETIC
F102 A2 00                  LDX     #0
F104 8A                     TXA

F105 95 00          INLP    STA     0,X             ;CLEAR HARDWARE REGISTERS AND RAM
F107 E8                     INX
F108 D0 FB                  BNE     INLP

F10A CA                     DEX                     ;$FF
F10B 9A                     TXS                     ;INIT STACK PTR

                            .IF     PRNT
                            STA     PRFLG           ;0 -- CLEAR PRINT FLAGS IF DEBUG
                            STA     PRDEP
                            STA     PRCNT
                            .ENDIF

F10C E6 B0                  INC     GAMNO           ;1

F10E AD 82 02               LDA     SWCHB           ;SAVE INITIAL PORT READING
F111 85 C2                  STA     OLDPBQ
F113 29 40                  AND     #$40
F115 85 EC                  STA     OLDPB6
F117 A2 30                  LDX     #$30
F119 86 18                  STX     AUDF1           ;FOR WIN SOUND
F11B E8                     INX
F11C 86 0A                  STX     CTRLPF          ;$31: 8 CLOCK BALL, REFLECT PLAYFIELD
F11E A2 04                  LDX     #4
F120 86 16                  STX     AUDC1

F122 20 82 F9               JSR     RESET           ;SET UP BOARD
```

```
                        ;                                   END OF FRAME -- RESTORE BOARD AFTER DISPLAY
F125                    FRAMLP
                                .IF     PAL                 ;TWO EXTRA LINES AT BOTTOM OF BOTTOM SQUARE IF PAL
                                JSR     WAIT3               ;DO 3 WSYNC'S AND RETURN
                                .ENDIF
                                .IF     PAL-1
F125 85 02                      STA     WSYNC               ;ANOTHER LINE AT BOTTOM OF BOARD IF NOT PAL
                                .ENDIF
F127 A2 25                      LDX     #OVRCON
F129 85 02                      STA     WSYNC
F12B 8E 96 02                   STX     TIM64T              ;SET TIMER FOR OVERSCAN (? LINES)
F12E A2 00                      LDX     #0
F130 86 0F                      STX     PF2                 ;CLEAR PLAYFIELD
F132 86 0E                      STX     PF1

F134 E6 C3                      INC     FRAME
F136 D0 08                      BNE     FRMLP2
F138 E6 EE                      INC     ATIM
F13A D0 04                      BNE     FRMLP2
F13C A9 00                      LDA     #0                  ;GO INTO ATTRACT
F13E 85 EF                      STA     GFLG2
F140                    FRMLP2
F140 A5 E5                      LDA     MOVFLG
F142 F0 0A                      BEQ     FRAM10              ;NO COMPUTER MOVE

                        ;                                   COMPUTER MOVE IS BEING DISPLAYED --- RESTORE
F144 18                         CLC
F145 20 CE F9                   JSR     JMPSET              ;CLEAR OUT JUMPED PIECES, IF ANY
F148 A6 E8                      LDX     CURSC
F14A A9 00                      LDA     #0
F14C 95 80                      STA     BOARD,X
F14E                    FRAM10
F14E A6 E9                      LDX     MOVE                ;"TO"
F150 A5 E4                      LDA     MOVVAL
F152 95 80                      STA     BOARD,X

F154 A6 AB                      LDX     CURSOR              ;"FROM"
F156 A5 A3                      LDA     PIECE
F158 95 80                      STA     BOARD,X             ;RESTORE CURRENT SQUARE

F15A A5 F0                      LDA     BLNKTM              ;STILL BLANKING AT END OF COMPUTER MOVE?
F15C D0 6F                      BNE     BLKJMP              ;YES.  SKIP

F15E AD 82 02                   LDA     SWCHB               ;NO.  SAVE NEW CONSOLE SWITCH PORT VALUE
F161 AA                         TAX                         ;IN OLDPBQ BUT DON'T CHANGE OLDPB6 YET.
F162 45 C2                      EOR     OLDPBQ
F164 86 C2                      STX     OLDPBQ
F166 10 62                      BPL     NOREV               ;NO CHANGE IN BIT 7 OF OLDPBQ
```

```
                        ;       RIGHT DIFFICULTY SWITCH MOVED MEANS TO REVERSE BOARD (SWITCH SIDES)
                        ;
                        ;                                       MOVE CURSOR, FROMB, TOSQRB
F168 A2 02                      LDX     #2
F16A A5 AD                      LDA     TOSQRB
F16C 30 07                      BMI     SW10            ;NO "TO" SQUARE
F16E                    SWLP
F16E A9 22                      LDA     #34
F170 38                         SEC
F171 F5 AB                      SBC     CURSOR,X
F173 95 AB                      STA     CURSOR,X
F175                    SW10
F175 CA                         DEX
F176 10 F6                      BPL     SWLP

F178 A9 22                      LDA     #34             ;CHANGE CURSC, MOVE
F17A 38                         SEC
F17B E5 E8                      SBC     CURSC
F17D 85 E8                      STA     CURSC
F17F A9 22                      LDA     #34
F181 38                         SEC
F182 E5 E9                      SBC     MOVE
F184 85 E9                      STA     MOVE

                        ;                                       SWITCH SIDES FOR EACH PIECE ON BOARD, PIECE
F186 A2 23                      LDX     #34+1
F188                    NORLP
F188 B5 80                      LDA     BOARD,X
F18A F0 04                      BEQ     NOR05
F18C 49 80                      EOR     #$80
F18E 95 80                      STA     BOARD,X
F190                    NOR05
F190 CA                         DEX
F191 10 F5                      BPL     NORLP

F193 A5 E4                      LDA     MOVVAL          ;SWITCH MOVVAL
F195 F0 04                      BEQ     NOR06
F197 49 80                      EOR     #$80
F199 85 E4                      STA     MOVVAL
F19B                    NOR06
                        ;                                       REVERSE BOARD
F19B A2 10                      LDX     #16
F19D A0 00                      LDY     #0              ;0 TO 16 (17 IS UNUSED), 18-34

F19F                    BSWLP
F19F B5 92                      LDA     BOARD+18,X
F1A1 48                         PHA
F1A2 B9 80 00                   LDA     BOARD,Y
F1A5 95 92                      STA     BOARD+18,X
F1A7 68                         PLA
F1A8 99 80 00                   STA     BOARD,Y
F1AB C8                         INY
F1AC CA                         DEX
F1AD 10 F0                      BPL     BSWLP

F1AF 20 14 FB                   JSR     SWIT            ;REVERSE ACTIVE
F1B2 A5 B1                      LDA     COLHUM          ;SWITCH COLORS
F1B4 A6 B2                      LDX     COLCMP
F1B6 86 B1                      STX     COLHUM
```

VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM

```
F1B8 85 B2                  STA     COLCMP

F1BA A5 A9                  LDA     HPIECE          ;SWITCH HPIECE & CPIECE
F1BC A6 AA                  LDX     CPIECE
F1BE 86 A9                  STX     HPIECE
F1C0 85 AA                  STA     CPIECE

F1C2 A5 AE                  LDA     HKING           ;SWITCH HKING & CKING
F1C4 A6 AF                  LDX     CKING
F1C6 86 AE                  STX     HKING
F1C8 85 AF                  STA     CKING

F1CA                NOREV

F1CA 20 44 F9               JSR     GSRST
```

```
                        ;                                       BEGINNING OF FRAME -- CHECK FOR SET UP MODE
F1CD                    BLKJMP
F1CD A9 03                      LDA     #3              ;B6=0 => DISABLE LATCHES, B1=1 => START VSYNC & VBLANK
F1CF AE 84 02           OVSLP   LDX     INTIM           ;WAIT FOR END OF OVERSCAN
F1D2 D0 FB                      BNE     OVSLP
                        ;                               START VERTICAL BLANK
F1D4 85 02                      STA     WSYNC
F1D6 85 00                      STA     VSYNC           ;START VERTICAL SYNC
F1D8 85 01                      STA     VBLANK
F1DA 86 15                      STX     AUDC0           ;NO SOUND
F1DC 85 0A                      STA     CTRLPF          ;STORE 3 -- SCORE, REFLECT PLAYFIELD
F1DE 85 02                      STA     WSYNC
F1E0 85 02                      STA     WSYNC
F1E2 A0 2C                      LDY     #VBLCON         ;37 LINES
F1E4 85 02                      STA     WSYNC
F1E6 86 00                      STX     VSYNC           ;END VERTICAL SYNC (X=0)
F1E8 8C 96 02                   STY     TIM64T

F1EB A5 F0                      LDA     BLNKTM
F1ED F0 03                      BEQ     NOBL
F1EF 4C D6 F7                   JMP     ATTR10          ;DO NOTHING
F1F2                    NOBL

                        ;                               CHECK FOR SETUP MODE (LEFT PRO/AM SWITCH UP (=1))
F1F2 A5 C2                      LDA     OLDPB0
F1F4 A6 EC                      LDX     OLDPB6
F1F6 D0 67                      BNE     STUP25
F1F8 29 40                      AND     #$40            ;PREVIOUSLY IN NORMAL MODE
F1FA D0 4E                      BNE     STUP10
                        ;
                        ;       STILL IN NORMAL MODE  (NORMAL -> NORMAL)
                        ;
F1FC A5 E6                      LDA     WINSAV
F1FE F0 13                      BEQ     COMP05          ;NO WIN
                        ;
                        ;       SOMEONE HAS WON -- OUTPUT WIN SOUND
                        ;
F200 A6 EF                      LDX     GFLG2
F202 F0 0C                      BEQ     WIN70           ;ATTRACT MODE
F204 C6 F1                      DEC     SNDTIM
F206 A6 F1                      LDX     SNDTIM
F208 86 EF                      STX     GFLG2           ;0=> DONE WITH SOUND, DO ATTRACT
F20A F0 02                      BEQ     WIN60
F20C A6 C3                      LDX     FRAME
F20E                    WIN60
F20E 86 1A                      STX     AUDV1
F210                    WIN70
F210 4C 66 F7                   JMP     CRSBLK


                        ;
                        ;       CHECK FOR COMPUTER OR HUMAN MOVE
                        ;
F213                    COMP05
F213 A6 B0                      LDX     GAMNO
F215 A5 F2                      LDA     FREEZE          ;FREEZE DISPLAY AFTER MOVE?
F217 F0 1F                      BEQ     COMP06          ;NO.  CONTINUE
F219 C6 F2                      DEC     FREEZE          ;FREEZE OVER YET?
F21B D0 F3                      BNE     WIN70           ;NO.  JMP CRSBLK
F21D E0 10                      CPX     #PLAY2
F21F D0 1B                      BNE     COMP07
```

```
F221 20 E8 F9                JSR     JMPWIN          ;CHECK FOR JUMP.  GAME OVER?
F224 F0 64                   BEQ     STUP50          ;YES.
F226 A5 AB                   LDA     CURSOR
F228 A6 E8                   LDX     CURSC           ;REVERSE COMPUTER AND HUMAN CURSORS
F22A 85 E8                   STA     CURSC
F22C 86 AB                   STX     CURSOR
F22E B5 80                   LDA     BOARD,X
F230 85 A3                   STA     PIECE
F232 20 E2 FA                JSR     TOSET           ;CLEAR TOSQRB, MOVFLG
F235 4C D6 F7                JMP     ATTR10

F238                 COMP06
F238 E0 10                   CPX     #PLAY2
F23A F0 04                   BEQ     COMP10          ;2-PLAYER GAME
F23C                 COMP07
F23C A6 BB                   LDX     ACTIVE
F23E 30 03                   BMI     CMPMVE          ;COMPUTER'S MOVE
F240                 COMP10
F240 4C A7 F5                JMP     NOCHCK          ;2-PLAYER GAME OR HUMAN'S MOVE
F243                 CMPMVE
F243 20 E8 F9                JSR     JMPWIN          ;SET UP MSTJMP, XMOVE, YMOVE,  ETC.
F246 F0 42                   BEQ     STUP50          ;GAME OVER -- GOTO ATTR10
F248 D0 43                   BNE     COMPUT          ;JMP    DEFINITELY DO COMPUTER'S MOVE

                     ;
                     ;       CHANGE TO SETUP MODE (NORMAL -> SETUP)
                     ;
F24A                 STUP10
F24A 85 EC                   STA     OLDPB6
F24C 20 E1 FE                JSR     MOVCLR          ;SET MOVE TO OFF BOARD SQUARE
F24F 20 F3 FA                JSR     TONORM          ;NOT SOFT ATTRACT (RETURNS Y=$FF)
F252 84 A4                   STY     JMPI            ;$FF -- NO JUMPS FOR COMPUTER
F254 20 E2 FA                JSR     TOSET           ;SET UP TOSQRB, MOVFLG, RETURN Y=0, EQ
F257 84 1A                   STY     AUDV1           ;CLEAR WIN SOUND
F259 84 E7                   STY     FROMTO
F25B 84 A5                   STY     MSTJMP          ;NO JUMP
F25D F0 04                   BEQ     STUP30          ;JMP

                     ;
                     ;       PREVIOUSLY IN SETUP MODE
                     ;
F25F                 STUP25
F25F 29 40                   AND     #$40
F261 85 EC                   STA     OLDPB6

F263                 STUP30
F263 20 FB FA                JSR     CRSTUP          ;SET UP ACTIVE, COLHUM,COLCMP

F266 A6 AB                   LDX     CURSOR
F268 B5 80                   LDA     BOARD,X         ;CURSOR SQUARE DETERMINES STARTING PLAYER AFTER SET UP
F26A F0 04                   BEQ     STUP35          ;  UNLESS EMPTY-- USE RIGHT DIFFICULTY SWITCH FROM CRSTUP
F26C 29 80                   AND     #$80
F26E 85 BB                   STA     ACTIVE
F270                 STUP35
F270 A2 00                   LDX     #0
F272 86 E6                   STX     WINSAV
F274 86 E5                   STX     MOVFLG          ;NO COMPUTER MOVE
F276 A6 EC                   LDX     OLDPB6
```

```
F278 D0 C6                      BNE     COMP10          ;GOTO NOCHCK IF STILL IN SETUP (NORMAL->SETUP OR SETUP->SETUP)
                        ;
                        ;       OUT OF SETUP (SETUP -> NORMAL)
                        ;
                                .IF     PRNT
                                JSR     PRBRD
                                .ENDIF

F27A 20 F3 FA                   JSR     TONORM          ;MAKE SURE WE'RE NOT IN ATTRACT MODE
F27D A6 B0                      LDX     GAMNO
F27F E0 10                      CPX     #PLAY2          ;TWO-PLAYER GAME?
F281 F0 04                      BEQ     STUP49          ;YES.
F283 A5 BB                      LDA     ACTIVE          ;NO.    COMPUTER PLAYER?
F285 30 03                      BMI     STUP50          ;YES.
F287                    STUP49
F287 20 E8 F9                   JSR     JMPWIN          ;CHECK FOR JUMPS, MOVES, WINS FOR HUMAN PLAYER
F28A                    STUP50
F28A 4C D6 F7                   JMP     ATTR10          ;SKIP THE REST UNTIL NEXT TIME
```

```
                        ;       CALL ALPHA-BETA PRUNING ROUTINE
                        ;       THIS IS A "RECURSIVE" TREE SEARCH.
                        ;       TO SAVE SPACE, THE STACK IS NOT USED.  INSTEAD, THE ROUTINE RETURNS
                        ;       TO ABRET1 IF DEPTH=1 AND OTHERWISE RETURNS TO ITSELF.
                        ;       EVENTUALLY, WE WANT TO CHECK TO SEE IF THERE IS ONLY ONE POSSIBLE MOVE
                        ;       FIRST, FOR FASTER RESPONSE.
                        ;
F28D                    COMPUT
F28D A2 01                      LDX     #1
F28F 86 BC                      STX     DEPTH           ;START AT DEPTH=1
F291 86 DE                      STX     ALPHAL+0
F293 86 DF                      STX     ALPHAL+1
F295 CA                         DEX                     ;0
F296 86 A6                      STX     MLTJMP          ;NOT MULTIPLE JUMP YET
F298 86 01                      STX     VBLANK          ;MAKE SURE VBLANK IS OFF
F29A 86 C4                      STX     PRNCNT          ;CLEAR PRUNE FLAG AND BEST MOVE COUNTER

F29C A9 80                      LDA     #$80            ;ALPHA(0) = ALPHA(1) = $8001 = -INFINITY
F29E 85 EB                      STA     ALPHAH+0
F2A0 85 EC                      STA     ALPHAH+1
F2A2 85 BF                      STA     FRSMOV          ;FIRST MOVE -- NO BETTER MOVE YET

                        ;
                        ;       IF RESET PRESSED THEN PICK OPENING MOVE AT RANDOM
                        ;
F2A4 A5 C2                      LDA     OLDPBQ
F2A6 6A                         ROR     A               ;RESET SWITCH PRESSED?
F2A7 B0 13                      BCS     COMPU2          ;NO.
F2A9 A5 C3                      LDA     FRAME           ;YES.  PICK FROM 7 OPENING MOVES
F2AB 29 07                      AND     #7              ;AT RANDOM (22-19) IS TWICE AS LIKELY AS THE OTHERS).
F2AD AA                         TAX
F2AE BD 6E FF                   LDA     FRSTAB,X
F2B1 85 C5                      STA     JMPLST
F2B3 29 3F                      AND     #$3F
F2B5 85 AC                      STA     FROMB
F2B7 85 E8                      STA     CURSC
F2B9 4C 5A F5                   JMP     NOJUMP          ;JMP
F2BC                    COMPU2
```

```
                        ;
                        ;       ALPHBETA -- ALPHA-BETA PRUNING ROUTINE  -- PUT INLINE TO SAVE A FEW BYTES
                        ;
                        ;       CALL BY JMP ALPHBETA, NOT JSR
                        ;       IF DEPTH = 1 THEN RETURN IS BY JMP ABRET1,
                        ;       OTHERWISE, JMP ABRET2.
                        ;
F2BC                    ALPHBETA
                        ;
                        ;       CHANGE COLORS IF TIMER UP
                        ;
                                .IF     PRNT
                                LDA     INPT5           ;IS RIGHT JOYSTICK BUTTON PRESSED?
                                BMI     AB05            ;NO.
                                JSR     PASK            ;YES.   INPUT NEW PRFLG FROM TI FOR DEBUG
                        AB05
                                .ENDIF

                        ;
                        ;       CHECK RESET SWITCH (DON'T CHECK GAME SELECT)
                        ;
F2BC AD 82 02                   LDA     SWCHB
F2BF 6A                         ROR     A               ;RESET PRESSED?
F2C0 B0 0B                      BCS     AB50            ;NO.
F2C2 2A                         ROL     A               ;SHIFT A BACK
F2C3 20 8D F5                   JSR     RSV2            ;YES.
F2C6 A9 1E                      LDA     #VBTIM
F2C8 85 F0                      STA     BLNKTM
F2CA 4C CA F1                   JMP     NOREV           ;GOTO CODE THAT DOES JSR GSRST


F2CD                    AB50
F2CD A5 A6                      LDA     MLTJMP          ;SAVE MLTJMP AND MSTJMP
F2CF 85 A8                      STA     MLTFLG
F2D1 A5 A5                      LDA     MSTJMP
F2D3 85 A7                      STA     MSTFLG

F2D5 A6 C0                      LDX     XMOVE           ;USE MOVE DISCOVERED BY JMPCHK
                        ;
                        ;       TRY NEXT SQUARE
                        ;
F2D7                    ALOOP1
F2D7 B5 80                      LDA     BOARD,X
F2D9 F0 04                      BEQ     CONT3J          ;EMPTY -- TRY NEXT SQUARE
F2DB 45 BB                      EOR     ACTIVE
F2DD 10 03                      BPL     ALPH05
F2DF                    CONT3J
F2DF 4C E4 F4                   JMP     CONT3           ;OPPONENT'S PIECE -- TRY NEXT SQUARE
F2E2 86 AC              ALPH05  STX     FROMB           ;ACTIVE -- SAVE SQUARE #
F2E4 20 22 FA                   JSR     LOADY           ;LOAD Y WITH 3 (OR 1 IF CMPCHK)
                        ;
                        ;       TRY NEXT INDEX
                        ;
F2E7                    ALPH10
F2E7 B5 80                      LDA     BOARD,X
F2E9 85 A3                      STA     PIECE           ;SAVE CONTENTS OF SQUARE (PIECE BEING MOVED)
F2EB 20 C9 FA                   JSR     ONBRD2          ;ADD MOVTAB OFFSET.  ON BOARD?
F2EE B0 40                      BCS     CONT2J          ;NO.   TRY NEXT OFFSET
```

```
F2F0 AA                  TAX
F2F1 A5 A5               LDA     MSTJMP
F2F3 D0 33               BNE     AJUMP           ;MUST JUMP
```

```
                        ;
                        ;       NO JUMP -- ORDINARY MOVE
                        ;
F2F5 B5 80              LDA     BOARD,X         ;IS "TO" SQUARE OCCUPIED?
F2F7 D0 37              BNE     CONT2J          ;YES.  TRY NEXT OFFSET

F2F9 86 AD              STX     TOSQRB          ;NO.  HAVE POTENTIAL MOVE
F2FB 84 BE              STY     INDEX           ;SAVE TO SQUARE AND INDEX
F2FD A5 A3              LDA     PIECE           ;LOAD PIECE AND CHECK FOR NEW KING
F2FF 20 BC FE           JSR     KNGCK2          ;CHANGES A AND PIECE COUNT IF NEW KING
F302 95 80              STA     BOARD,X         ;CHANGE TOSQR TO FROM SQUARE
F304 A9 00              LDA     #0              ;CLEAR FROM SQUARE
F306 A6 AC              LDX     FROMB
F308 95 80              STA     BOARD,X

F30A 20 CE FB           JSR     TERMNODE        ;CHECK FOR TERMINAL NODE
F30D B0 0C              BCS     TER             ;YES.  -- TERMINAL

                        ;
                        ;       NOT TERMINAL NODE -- SAVE VARS.  FOR THIS LEVEL
                        ;
F30F A5 A3              LDA     PIECE
F311 29 30              AND     #$30            ;BIT 5-4 INDICATES KING OR CHECKER
F313 05 BE              ORA     INDEX           ;BIT 1-0 IS MOVTAB INDEX (MSTJMP AND MLTJMP ARE 0)
F315 20 AF FE           JSR     SAVE1           ;SAVE OTHER, FROMT, INC DEPTH.  RETURN X=DEPTH
F318 4C 76 F3           JMP     ALPJMP

                        ;
                        ;       TER -- ORDINARY MOVE -- NOT JUMP -- TERMINAL NODE
                        ;
F31B                    TER
F31B 20 14 FB                   JSR     SWIT
                        ;
                        ;       TAKE MOVE BACK
F31E 20 62 FB                   JSR     RESTOR
F321 A9 00                      LDA     #0              ;MAKE SURE MSTJMP IS CLEAR BECAUSE IF AT MAXMAX DEPTH,.
                        ;                               THEN TERMNODE MIGHT HAVE DETECTED JUMP.
F323 85 A5                      STA     MSTJMP
F325 4C 4A F4                   JMP     FINSH3          ;CHECK NEW STATIC VALUE, POSSIBLY MODIFY ALPHA
```

```
                        ;
                        ;       AJUMP -- MUST JUMP
                        ;
F328                    AJUMP
F328 B5 80                      LDA     BOARD,X
F32A F0 04                      BEQ     CONT2J          ;EMPTY -- TRY NEXT OFFSET
F32C 45 BB                      EOR     ACTIVE
F32E 30 03                      BMI     AJ05
F330                    CONT2J
F330 4C D2 F4                   JMP     CONT2           ;ACTIVE PLAYER -- CAN'T JUMP OWN PIECE
F333                    AJ05
F333 86 BD                      STX     JMPSQR          ;SAVE SQUARE WHICH WE ARE JUMPING OVER
F335 20 C9 FA                   JSR     ONBRD2          ;ADD MOVTAB OFFSET.  OFF BOARD?
F338 B0 F6                      BCS     CONT2J          ;YES.  TRY NEXT OFFSET
F33A AA                         TAX
F33B B5 80                      LDA     BOARD,X
F33D D0 F1                      BNE     CONT2J          ;TO SQUARE IS OCCUPIED -- TRY NEXT OFFSET

F33F 86 AD                      STX     TOSQRB          ;SUCCESS -- SAVE SQUARE
F341 84 BE                      STY     INDEX
F343 A5 A3                      LDA     PIECE
F345 20 DF FF                   JSR     KNGCHK          ;IF NEW KING THEN SET CARRY, INC PIECE COUNT & MODIFY MSTJMP
F348 95 80                      STA     BOARD,X         ;STORE IN TO SQUARE
F34A A0 00                      LDY     #0              ;CLEAR FROM SQUARE
F34C A6 AC                      LDX     FROMB
F34E 94 80                      STY     BOARD,X
F350 A6 BD                      LDX     JMPSQR
F352 B5 80                      LDA     BOARD,X
F354 0A                         ASL     A               ;SAVE JUMPED PIECE IN B7-6
F355 0A                         ASL     A
F356 05 BD                      ORA     JMPSQR          ;COMBINE WITH JUMP SQUARE
F358 85 BD                      STA     JMPSQR
F35A 94 80                      STY     BOARD,X         ;CLEAR JUMP SQUARE
F35C 20 51 FE                   JSR     CHGCNT          ;CHANGE PIECE COUNTS



F35F A5 A5                      LDA     MSTJMP          ;MUST JUMP?
F361 F0 07                      BEQ     MJ05            ;NO -- NEW KING MUST STOP -- CHECK FOR NEW JUMP
F363 A6 AD                      LDX     TOSQRB
F365 20 2D FA                   JSR     MOVCHK
F368 90 22                      BCC     MJ08            ;ANOTHER JUMP FOR THIS PIECE
```

```
                        ;
                        ;       NEW KING OR END OF JUMP
                        ;
F36A                    MJ05
F36A A9 00                      LDA     #0              ;NOT MULTIPLE JUMP NEXT TIME
F36C 85 A6                      STA     MLTJMP

F36E 20 CE FB                   JSR     TERMNODE        ;SWITCH PLAYERS -- CHECK FOR TERMINAL
                        ;                               SET UP XMOVE, MSTJMP FOR NEXT LEVEL
F371 B0 14                      BCS     MJ07            ;TERMINAL
                        ;                               NOT TERMINAL
F373 20 9F FE                   JSR     JMPSAV          ;INC DEPTH & SAVE VARS   RETURN X=NEW DEPTH
F376                    ALPJMP
F376 B5 DC                      LDA     ALPHAL-2,X
F378 95 DE                      STA     ALPHAL,X
F37A B5 E9                      LDA     ALPHAH-2,X
F37C 95 EB                      STA     ALPHAH,X

F37E A9 00                      LDA     #0
F380 85 BF                      STA     FRSMOV          ;NOT FIRST MOVE
F382 85 A6                      STA     MLTJMP


F384 4C BC F2                   JMP     ALPHBETA        ;RECURSE

F387                    MJ07
F387 20 14 FB                   JSR     SWIT            ;TERMINAL -- SWITCH ACTIVE BACK
F38A B0 0F                      BCS     MJ15            ;JMP

                        ;
                        ;       MULTIPLE JUMP -- MSTJMP ALREADY SET
                        ;
F38C                    MJ08
F38C A9 08                      LDA     #MLT
F38E 85 A6                      STA     MLTJMP
F390 20 14 FB                   JSR     SWIT
F393 20 E5 FB                   JSR     ALP200          ;CHECK FOR MAXMAXDEPTH
F396 20 14 FB                   JSR     SWIT
F399 90 0D                      BCC     MJ20            ;OK - KEEP GOING
                        ;
                        ;       STACK FULL -- MUST STOP -- UNDO JUMP
                        ;
F39B                    MJ15
F39B A5 BD                      LDA     JMPSQR
F39D 29 3F                      AND     #$3F
F39F AA                         TAX
F3A0 A5 BD                      LDA     JMPSQR

F3A2 20 3C FB                   JSR     RESTR5          ;RESTORES MSTJMP, MLTJMP, ETC.

F3A5 4C 4A F4                   JMP     FINSH3          ;CHECK NEW STATIC VALUE
```

```
                        ;
                        ;       SAVE VARS -- MULTIPLE JUMP NEXT TIME
                        ;
F3A8                    MJ20
F3A8 20 9F FE                   JSR     JMPSAV          ;INC DEPTH & SAVE VARS (MSTFLG, MLTFLG, ETC.)

F3AB A5 BF                      LDA     FRSMOV
F3AD 29 80                      AND     #$80            ;FIRST MOVE STAYS THE SAME, EXCEPT NO BETTER MOVE
F3AF 85 BF                      STA     FRSMOV

F3B1 B5 DD                      LDA     ALPHAL-1,X      ;USE SAME ALPHA
F3B3 95 DE                      STA     ALPHAL,X
F3B5 B5 EA                      LDA     ALPHAH-1,X
F3B7 95 EB                      STA     ALPHAH,X

F3B9 B5 DC                      LDA     ALPHAL-2,X      ;ALSO SAME BETA (PREVIOUS ALPHA)
F3BB 95 DD                      STA     ALPHAL-1,X
F3BD B5 E9                      LDA     ALPHAH-2,X
F3BF 95 EA                      STA     ALPHAH-1,X

F3C1 4C BC F2                   JMP     ALPHBETA
```

```
                        ;
                        ;       RETURN FROM RECURSIVE ALPHBETA CALL
                        ;
                        ;
                        ;       RESTORE VARS
                        ;
F3C4                    ABRET2
F3C4 C6 BC                      DEC     DEPTH
F3C6 A6 BC                      LDX     DEPTH

F3C8 B5 C7                      LDA     FROMT-1,X
F3CA 29 3F                      AND     #$3F
F3CC 85 AC                      STA     FROMB
F3CE B5 D2                      LDA     OTHER-1,X
F3D0 29 08                      AND     #MLT
F3D2 85 A8                      STA     MLTFLG
F3D4 B5 D2                      LDA     OTHER-1,X
F3D6 29 04                      AND     #MST
F3D8 85 A7                      STA     MSTFLG
F3DA B5 D2                      LDA     OTHER-1,X
F3DC 29 03                      AND     #3
F3DE 85 BE                      STA     INDEX

F3E0 A5 A7                      LDA     MSTFLG          ;WAS MOVE A JMP?
F3E2 D0 29                      BNE     WASJ            ;YES.
                        ;
                        ;       NOT A JUMP
                        ;
F3E4 20 14 FB                   JSR     SWIT            ;SWITCH ACTIVE PLAYER
F3E7 A5 A8                      LDA     MLTFLG          ;NO.  NOT JUMP
F3E9 85 A6                      STA     MLTJMP
F3EB A5 A7                      LDA     MSTFLG
F3ED 85 A5                      STA     MSTJMP

F3EF B5 D2                      LDA     OTHER-1,X       ;SET UP PIECE
F3F1 29 30                      AND     #$30
F3F3 05 BB                      ORA     ACTIVE
F3F5 85 A3                      STA     PIECE

F3F7 B5 C7                      LDA     FROMT-1,X
F3F9 29 C0                      AND     #$C0
F3FB 85 BF                      STA     FRSMOV

F3FD A5 AC                      LDA     FROMB
F3FF A4 BE                      LDY     INDEX
F401 18                         CLC
F402 79 66 FF                   ADC     MOVTAB,Y
F405 85 AD                      STA     TOSQRB
F407 20 62 FB                   JSR     RESTOR          ;RESTORE SOME VARS & BOARD
F40A 4C 19 F4                   JMP     PRNJMP          ;CLEAR PRUNE FLAG & CHECK NEW VALUE AGAINST ALPHA

                        ;
                        ;       WAS A JUMP
                        ;
F40D                    WASJ
F40D A4 A6                      LDY     MLTJMP
F40F D0 10                      BNE     WASJ10
                        ;
                        ;       WAS NOT A MULTIPLE JUMP
                        ;
```

```
F411 20 14 FB                JSR     SWIT            ;SWITCH ACTIVE PLAYER
F414 B5 C7                   LDA     FROMT-1,X
F416 20 1B FB                JSR     RESTR6          ;SET UP FRSMOV,PIECE,MSTJMP,MLTJMP, RESTORE BOARD
F419                 PRNJMP
F419 06 C4                   ASL     PRNCNT          ;CLEAR PRUNE FLAG WITHOUT DISTURBING COUNTER
F41B 18                      CLC
F41C 66 C4                   ROR     PRNCNT
F41E 4C 4A F4                JMP     FINSH3

                     ;
                     ;       NEXT LEVEL CONTINUED MULTIPLE JUMP -- MOVE BETA & NEW ALPHA BACK
                     ;
F421                 WASJ10
                             .IF     PRNT
                             LDA     PRFLG
                             AND     #$80            ;FLAG SET?
                             BEQ     SKIP1           ;NO.
                             TXA                     ;YES.  SAVE X AND PRINT "MULTIPLE JUMP" MESSAGE
                             PHA
                             LDA     #MJ
                             JSR     BLECCH
                             PLA                     ;RESTORE X
                             TAX
                     SKIP1
                             .ENDIF

F421 B5 DE                   LDA     ALPHAL,X                ;MOVE BETA
F423 95 DD                   STA     ALPHAL-1,X
F425 B5 EB                   LDA     ALPHAH,X
F427 95 EA                   STA     ALPHAH-1,X

F429 B5 EC                   LDA     ALPHAH+1,X              ;MOVE ALPHA
F42B 95 EB                   STA     ALPHAH,X
F42D B5 DF                   LDA     ALPHAL+1,X
F42F 95 DE                   STA     ALPHAL,X

F431 A5 C4                   LDA     PRNCNT          ;PRUNE?
F433 10 06                   BPL     NEW50           ;NO.

F435 20 1F FB                JSR     RESTR4          ;YES.  LEAVE FRSMOV AS IS.  RESTORE OTHER JUNK
F438 4C 0A F5                JMP     ABD2            ;GOTO PRUNE
F43B                 NEW50

F43B 24 BF                   BIT     FRSMOV          ;FIRST MOVE? (NEXT LEVEL)
F43D 10 05                   BPL     SKIP2           ;NO.  BETTER MOVE?
F43F 50 03                   BVC     SKIP2           ;NO.  DON'T SAVE MOVE BUT RESTORE JUNK
                     ;
                     ;       MULTIPLE JUMP -- FIRST MOVE & BETTER MOVE
                     ;
                     ;       KEEP FRSMOV & BETTER MOVE FROM PREVIOUS LEVEL (BOTH SET)
                     ;
F441                 WASJ17
F441 20 6A FE                JSR     JMPSV2

                             .IF     PRNT
                             LDA     PRFLG
                             AND     #$40
                             BEQ     SKIP2
                             LDA     #MJBA           ;"MULTIPLE JUMP - FIRST MOVE & BETTER ALPHA"
                             JSR     BLECCH
```

```
                                .ENDIF

F444                    SKIP2
F444 20 1F FB                   JSR     RESTR4          ;SET UP PIECE,MSTJMP,MLTJMP,RESTORE BOARD
F447 4C D0 F4                   JMP     CONT1           ;SKIP ALPHA CHECKING AND DO NEXT INDEX
```

Moore Business Forms, Inc.

```
                        ;
                        ;       CHECK NEW VALUE AGAINST ALPHA
                        ;       MAY BE STATIC VALUE OR VALUE PASSED BACK BY CALL TO NEXT DEPTH.
                        ;
                        ;       HUMAN PASSES UP -C/H, COMPUTER TRIES TO MAXIMIZE +C/H (MAKE H SMALLER RELATIVE TO C).
                        ;       COMPUTER PASSES UP +C/H, HUMAN TRIES TO MAXIMIZE -C/H (MAKE H LARGER RELATIVE TO C)
                        ;
F44A                    FINSH3
                        ;
                        ;       IF -ALPHA(DEPTH+1) > ALPHA(DEPTH) THEN HAVE BETTER ALPHA.
                        ;       I.E.  IF ALPHA(DEPTH) + ALPHA(DEPTH+1) < 0 THEN HAVE BETTER ALPHA.
                        ;
F44A A6 BC                      LDX     DEPTH

F44C 24 BF                      BIT     FRSMOV          ;FIRST MOVE?
F44E 10 06                      BPL     NEW10           ;NO.
F450 F6 DF                      INC     ALPHAL+1,X      ;YES.   INCREASE -ALPHA BY 1 (TO MAKE RANDOMNESS WORK)
F452 D0 02                      BNE     NEW10
F454 F6 EC                      INC     ALPHAH+1,X      ;IF LSB WENT TO 0 THEN INC MSB
F456                    NEW10

F456 B5 DE                      LDA     ALPHAL,X
F458 18                         CLC
F459 75 DF                      ADC     ALPHAL+1,X
F45B 85 B3                      STA     T0              ;SAVE LSB
F45D B5 EB                      LDA     ALPHAH,X
F45F 75 EC                      ADC     ALPHAH+1,X
F461 50 02                      BVC     FH5             ;OVERFLOW?
F463 B5 EB                      LDA     ALPHAH,X        ;YES.   OVERFLOW -- LOAD SIGN
F465                    FH5
F465 30 34                      BMI     FH7             ;NEGATIVE => HAVE BETTER ALPHA
                        ;
                        ;       NEW VALUE IS THE SAME AS OLD ALPHA OR WORSE
                        ;
F467 70 2F                      BVS     FH6             ;OVERFLOW SET -- NOT 0 -- WORSE
F469 D0 2D                      BNE     FH6             ;MSB NOT 0 -- WORSE
F46B 24 BF                      BIT     FRSMOV          ;FIRST MOVE?
F46D 10 29                      BPL     FH6             ;NO.   FORGET THIS MOVE
F46F A5 B3                      LDA     T0              ;NEW VALUE = ALPHA?
F471 D0 25                      BNE     FH6             ;NO.   FORGET THIS MOVE

                                .IF     PRNT
                                LDA     PRFLG
                                AND     #$02
                                BEQ     SKIP7
                                LDY     DEPTH
                                CPY     PRDEP
                                BCS     SKIP7           ;DON'T PRINT IF DEPTH>=PRDEP

                                LDA     #RAN            ;"RANDOM"
                                JSR     OUTAL3
                                LDX     DEPTH
                        SKIP7
                                .ENDIF

F473 A5 C3                      LDA     FRAME           ;SUM IS 0  -- USE NEW VALUE AT "RANDOM"
F475 A8                         TAY                     ;MOVE OLD FRAME VALUE TO Y AND COMPUTE NEW VALUE
F476 D0 04                      BNE     FRAG
F478 A9 62                      LDA     #$62            ;SET TO NEW VALUE IF 0
F47A 85 C3                      STA     FRAME
```

```
F47C                    FRAQ
F47C 2A                         ROL     A               ;B7-> CARRY
F47D 26 C3                      ROL     FRAME           ;B7 -> B0
F47F 2A                         ROL     A               ;OLD B6 -> CARRY
F480 2A                         ROL     A               ;OLD B6-> B0
F481 29 01                      AND     #1
F483 45 C3                      EOR     FRAME
F485 85 C3                      STA     FRAME

F487 A5 C4                      LDA     PRNCNT          ;LOAD COUNTER
F489 29 7F                      AND     #$7F            ;MASK OFF PRUNE FLAG
F48B C9 05                      CMP     #5              ;EQUAL TO 5?
F48D B0 02                      BCS     NEW40           ;YES.  LIMIT TO 5
F48F E6 C4                      INC     PRNCNT          ;NO.  INCREMENT EQUAL MOVE COUNTER
F491                    NEW40
F491 AA                         TAX
F492 98                         TYA
F493 DD EA FE                   CMP     RANTAB,X        ;IS RANDOM NUMBER < 1/2, 1/3, 1/4 ETC?
F496 90 1F                      BCC     FHQ             ;YES.  KEEP MOVE

                        ;
                        ;       NEW VALUE WAS THE SAME AND WE ARE NOT USING THIS MOVE OR
                        ;       NEW VALUE WAS WORSE THAN THE OLD ALPHA.
                        ;
F498                    FH6
                                .IF     PRNT
                                LDA     PRFLG
                                AND     #$20
                                BEQ     SKIP3
                                LDA     #WA             ;"WORSE ALPHA"
                                JSR     BLECCH
                        SKIP3
                                .ENDIF
F498 4C D0 F4                   JMP     CONT1
                        ;
                        ;       IF ALPHA >= BETA THEN PRUNE TREE -- OPPONENT WOULD NEVER LET US PICK THIS MOVE
                        ;       I.E.  IF ALPHA(DEPTH) + ALPHA(DEPTH-1) >= 0 THEN PRUNE TREE
                        ;       BETA(DEPTH) = -ALPHA(DEPTH-1)
                        ;
F49B                    FH7
                        ;
                        ;       NEGATE ALPHA+1 AND MOVE TO ALPHA (X=DEPTH ON INPUT)
                        ;
F49B A9 00                      LDA     #0              ;NEGATE AND MOVE NEW VALUE TO ALPHA
F49D 38                         SEC
F49E F5 DF                      SBC     ALPHAL+1,X
F4A0 95 DE                      STA     ALPHAL,X
F4A2 A9 00                      LDA     #0
F4A4 F5 EC                      SBC     ALPHAH+1,X
F4A6 95 EB                      STA     ALPHAH,X

F4A8 B5 DE                      LDA     ALPHAL,X
F4AA 18                         CLC
F4AB 75 DD                      ADC     ALPHAL-1,X
F4AD B5 EB                      LDA     ALPHAH,X
F4AF 75 EA                      ADC     ALPHAH-1,X
F4B1 50 02                      BVC     FH20
F4B3 B5 EB                      LDA     ALPHAH,X        ;OVERFLOW -- LOAD SIGN
F4B5                    FH20
F4B5 10 53                      BPL     ABD2            ;MOVE REJECTED => PRUNE
```

```
                        ;
                        ;       BEST MOVE SO FAR -- SAVE IN JMPLST IF FIRST MOVE.
                        ;       RESULT OF ADDITION WAS NEGATIVE
                        ;
F4B7                    FHQ
F4B7 A5 BF                      LDA     FRSMOV
F4B9 09 40                      ORA     #$40            ;INDICATE BETTER MOVE
F4BB 85 BF                      STA     FRSMOV


                                .IF     PRNT
                                BMI     FHBLEC          ;BRANCH IF FIRST MOVE
                                LDA     PRFLG
                                AND     #$10
                                BEQ     SKIP4
                                LDA     #BANFM          ;"BETTER ALPHA - NOT FIRST MOVE"
                                JSR     BLECCH
                        SKIP4
                                JMP     CONT1
                                .ENDIF
F4BD 10 11                      BPL     CONT1           ;IF NOT DEBUG THEN BRANCH IF NOT FIRST MOVE
F4BF                    FHBLEC
F4BF A9 00                      LDA     #0              ;FIRST MOVE => CLEAR PRUNE FLAG & COUNTER
F4C1 85 C4                      STA     PRNCNT
F4C3 A9 FF                      LDA     #$FF
F4C5 A6 A5                      LDX     MSTJMP
F4C7 F0 02                      BEQ     FH21
F4C9 A5 BC                      LDA     DEPTH           ;JUMP
F4CB                    FH21
F4CB 85 A4                      STA     JMPI

F4CD 20 6A FE                   JSR     JMPSV2          ;SAVE MOVE IN JMPLST

                                .IF     PRNT
                                LDA     PRFLG
                                AND     #$08
                                BEQ     SKIP5
                                LDA     #BAFM           ;"BETTER ALPHA - FIRST MOVE"
                                JSR     BLECCH
                        SKIP5
                                .ENDIF
```

```
                        ;
                        ;       TRY NEXT INDEX
                        ;
F4D0                    CONT1
F4D0 A4 BE                      LDY     INDEX
                        ;
                        ;       TRY NEXT INDEX -- Y=INDEX
F4D2                    CONT2
F4D2 A6 AC                      LDX     FROMB
F4D4 88                         DEY
F4D5 30 0D                      BMI     CONT3           ;DONE WITH THIS SQUARE
F4D7 C0 01                      CPY     #1
F4D9 D0 06                      BNE     ALJ
F4DB B5 80                      LDA     BOARD,X
F4DD C9 10                      CMP     #HUMCHK         ;STOP IF HUMAN CHECKER
F4DF F0 03                      BEQ     CONT3
F4E1                    ALJ
F4E1 4C E7 F2                   JMP     ALPH10          ;CONTINUE

                        ;
                        ;       TRY NEXT SQUARE -- X=SQUARE
                        ;
F4E4                    CONT3
F4E4 C6 C1                      DEC     TIMOUT          ;TIME UP?
F4E6 D0 18                      BNE     AB30            ;NO.
F4E8 AD 82 02                   LDA     SWCHB
F4EB 29 08                      AND     #8              ;COLOR TV?
F4ED D0 07                      BNE     AB10            ;YES.
F4EF A5 BC                      LDA     DEPTH           ;NO.  B/W -- USE DEPTH TO SET LUMINANCE
F4F1 0A                         ASL     A
F4F2 29 06                      AND     #6
F4F4 10 08                      BPL     AB20            ;JMP
F4F6                    AB10
F4F6 A5 BC                      LDA     DEPTH           ;COLOR TV
F4F8 6A                         ROR     A
F4F9 6A                         ROR     A
F4FA 6A                         ROR     A
F4FB 6A                         ROR     A
F4FC 09 02                      ORA     #2              ;SET LUMINANCE
F4FE                    AB20
F4FE 85 09                      STA     COLUBK
F500                    AB30

F500 A5 A6                      LDA     MLTJMP
F502 D0 11                      BNE     ABDONE          ;MULTIPLE JUMP => DONE (ONLY TRY ONE SQUARE)

F504 CA                         DEX
F505 30 0E                      BMI     ABDONE          ;SKIP
F507 4C D7 F2                   JMP     ALOOP1          ;CONTINUE
                        ;
                        ;       DONE WITH ALL SQUARES -- RETURN
F50A                    ABD2
                                .IF     PRNT            ;WE COME HERE FOR PRUNING
                                LDA     PRFLG
                                AND     #$04
                                BEQ     SKIP6
                                LDA     #PRUNE          ;"PRUNE"
                                JSR     BLECCH
                        SKIP6
                                .ENDIF
```

```
F50A 06 C4                      ASL     PRNCNT          ;SET PRUNE FLAG (FOR MULTIPLE JUMPS)
F50C 38                         SEC
F50D 66 C4                      ROR     PRNCNT
F50F A5 BF                      LDA     FRSMOV          ;NOT BETTER MOVE
F511 29 80                      AND     #$80
F513 85 BF                      STA     FRSMOV
F515                    ABDONE
F515 A5 BC                      LDA     DEPTH
F517 C9 01                      CMP     #1
F519 F0 03                      BEQ     ABRET1          ;RETURN FOR DEPTH = 1
F51B 4C C4 F3                   JMP     ABRET2          ;RETURN FOR DEPTH > 1
```

```
F51E                    ABRET1
F51E A5 C5                      LDA     JMPLST
F520 29 3F                      AND     #$3F            ;START SQUARE
F522 85 AC                      STA     FROMB
F524 85 E8                      STA     CURSC
F526 A6 A4                      LDX     JMPI
F528 30 30                      BMI     NOJUMP          ;NO JUMPS

                        ;
                        ;       JUMPS
                        ;
F52A C6 A4                      DEC     JMPI            ;CHANGE 1-9 TO 0-8 (JMPLST INDEX)
F52C 85 B3                      STA     TO
F52E A9 00                      LDA     #0              ;JUMPS
F530 85 B4                      STA     T1              ;START WITH JMPLST+0
F532 06 C5                      ASL     JMPLST
F534 2A                         ROL     A
F535 06 C5                      ASL     JMPLST          ;B7-6 = INDEX INTO MOVTAB
F537 2A                         ROL     A
F538 A8                         TAY
F539 20 80 FB                   JSR     JUMP1           ;SET UP JMPLST FOR FIRST JUMP
F53C B0 0E                      BCS     JTHRU           ;DONE
F53E A5 C7                      LDA     JMPLST+2        ;MOVE 3RD BYTE TO END OF JMPLST TO SAVE IT
F540 48                         PHA
F541 A5 C6                      LDA     JMPLST+1        ;2ND BYTE CONTAINS 4 MOVTAB INDEXES
F543 20 B5 FB                   JSR     JUMP4           ;DO 4 JUMPS
F546 68                         PLA
F547 B0 03                      BCS     JTHRU           ;BRANCH IF THROUGH
F549 20 B5 FB                   JSR     JUMP4

F54C                    JTHRU
F54C A6 AC                      LDX     FROMB
F54E B5 80                      LDA     BOARD,X
F550 48                         PHA
F551 A9 00                      LDA     #0              ;CLEAR FROM SQUARE
F553 95 80                      STA     BOARD,X
F555 A5 B3                      LDA     TO              ;TO SQUARE
F557 4C 6E F5                   JMP     JTHRU2

                        ;
                        ;       NO JUMP -- JUST MAKE 1 MOVE   A=START SQUARE FROM JMPLST
                        ;
F55A                    NOJUMP
F55A AA                         TAX
F55B B5 80                      LDA     BOARD,X
F55D 48                         PHA
F55E A9 00                      LDA     #0
F560 95 80                      STA     BOARD,X
F562 06 C5                      ASL     JMPLST
F564 2A                         ROL     A
F565 06 C5                      ASL     JMPLST
F567 2A                         ROL     A
F568 A8                         TAY                     ;OFFSET INDEX
F569 8A                         TXA
F56A 18                         CLC
F56B 79 66 FF                   ADC     MOVTAB,Y        ;TO SQUARE
F56E                    JTHRU2
F56E AA                         TAX
F56F 85 AD                      STA     TOSQRB
```

```
F571 85 E9              STA     MOVE
F573 68                 PLA                     ;PIECE BEING MOVED
F574 20 BC FE           JSR     KNGCK2          ;IF NEW KING THEN CHANGE AND INCREMENT PIECE COUNT
F577 95 80              STA     BOARD,X
F579 85 E4              STA     MOVVAL
```

```
                        ;
                        ;       RESTORE VARIABLES -- END OF COMPUTER MOVE
                        ;
F57B 20 8A F5                   JSR     RSV

F57E A6 AB                      LDX     CURSOR          ;SET UP PIECE IN CASE CURSOR'S PIECE WAS JUMPED
F580 B5 80                      LDA     BOARD,X
F582 85 A3                      STA     PIECE

F584 20 E8 F9                   JSR     JMPWIN          ;CHECK FOR WIN
F587 4C D6 F7                   JMP     ATTR10


F58A                    RSV
F58A AD 82 02                   LDA     SWCHB
F58D                    RSV2
F58D 85 C2                      STA     OLDPBQ
F58F 29 40                      AND     #$40
F591 85 EC                      STA     OLDPB6

F593 20 F3 FA                   JSR     TONORM          ;MAKE SURE IN NORMAL MODE, Y<-$FF
F596 C8                         INY                     ;0
F597 84 A6                      STY     MLTJMP          ;NOT MULTIPLE JUMP CONTINUATION
F599 84 BB                      STY     ACTIVE          ;HUMAN IS ACTIVE
F59B 84 E7                      STY     FROMTO          ;"FROM"
F59D 84 ED                      STY     ILEGAL          ;NOT ILLEGAL MOVE
F59F 84 E6                      STY     WINSAV          ;NO WIN
F5A1 84 F2                      STY     FREEZE          ;NO FREEZE
F5A3 C8                         INY                     ;1
F5A4 84 E5                      STY     MOVFLG          ;DISPLAY COMPUTER'S MOVE

                                .IF     PRNT
                                JSR     PRMOVE          ;PRINT MOVE ON TI
                                .ENDIF
F5A6 60                         RTS
```

```
F5A7                    NOCHCK
                        ;                                       CHECK JOYSTICK BUTTON -- SET UP MODE?
                                .IF     PRNT
                                LDA     INPT5           ;IS RIGHT JOYSTICK BUTTON PUSHED?
                                BMI     JOY2            ;NO.
                                JSR     PASK            ;YES.   INPUT PRFLG FROM TI
                        JOY2
                                .ENDIF
F5A7 A5 EC                      LDA     OLDPB6
F5A9 F0 51                      BEQ     JOY11
F5AB A5 3C                      LDA     INPT4           ;SETUP MODE:  CHECK PUSHBUTTON
F5AD 10 04                      BPL     JOY3

                        ;
                        ;       BUTTON NOT PRESSED
                        ;
F5AF 18                         CLC                     ;ALWAYS USE P0'S JOYSTICK IN SETUP
F5B0 4C D7 F6                   JMP     BNEQ            ;UNCOND JMP
F5B3                    JOY3
F5B3 A5 C3                      LDA     FRAME           ;BUTTON IS PUSHED
F5B5 29 1F                      AND     #$1F            ;MOVE ONLY ON CERTAIN FRAMES
F5B7 D0 57                      BNE     JOY111          ;GOTO CRSBLK
F5B9 A5 A3                      LDA     PIECE           ;ALTERNATE BETWEEN COMPUTER & HUMAN CHECKER, COMPUTER & HUMAN KING, & SPACE
F5BB A2 04                      LDX     #4
F5BD                    JOYLP
F5BD DD 4E FF                   CMP     NXTAB,X         ;FIND CURRENT SQUARE CONTENTS IN TABLE
F5C0 F0 03                      BEQ     JOY9            ;FOUND
F5C2 CA                         DEX                     ;NOT FOUND
F5C3 10 F8                      BPL     JOYLP           ;CONTINUE

                                .IF     PRNT
                                LDA     #$FF
                                STA     PRFLG
                                LDA     #JLERR
                                JSR     OUTMSG          ;PRINT ERROR MESSAGE ON TI
                                JMP     *               ;ERROR (SHOULD NEVER OCCUR)
                                .ENDIF

F5C5                    JOY9
F5C5 A5 A9                      LDA     HPIECE          ;MODIFY PIECE COUNT TO REFLECT CHANGE
F5C7 18                         CLC
F5C8 7D 3E FF                   ADC     HTAB,X
F5CB 85 A9                      STA     HPIECE
F5CD A5 AA                      LDA     CPIECE
F5CF 18                         CLC
F5D0 7D 40 FF                   ADC     CTAB,X
F5D3 85 AA                      STA     CPIECE

F5D5 A5 AE                      LDA     HKING           ;MODIFY KING COUNT TO REFLECT CHANGE
F5D7 18                         CLC
F5D8 7D 43 FF                   ADC     HKTAB,X
F5DB 85 AE                      STA     HKING
F5DD A5 AF                      LDA     CKING
F5DF 18                         CLC
F5E0 7D 45 FF                   ADC     CKTAB,X
F5E3 85 AF                      STA     CKING

F5E5 CA                         DEX                     ;TAKE PREVIOUS ENTRY FROM TABLE
```

```
F5E6 10 02                  BPL     JOY10
F5E8 A2 04                  LDX     #4              ;0 GOES TO 4
F5EA                JOY10
F5EA BD 4E FF               LDA     NXTAB,X         ;LOAD NEXT SQUARE VALUE
F5ED A6 AB                  LDX     CURSOR
F5EF 20 BC FE               JSR     KNGCK2          ;MUST HAVE KING IF LAST ROW
F5F2 85 A3                  STA     PIECE
F5F4 95 80                  STA     BOARD,X
F5F6 20 F3 FA               JSR     TONORM
F5F9 4C CA F6               JMP     JOY81
```

```
                        ;
F5FC                    JOY11                           CHECK JOYSTICK BUTTON -- NOT SET UP MODE
F5FC A5 BB                      LDA     ACTIVE
F5FE 0A                         ASL     A
F5FF 2A                         ROL     A
F600 A8                         TAY                     ;Y=0 IF ACTIVE=0, 1 IF ACTIVE =$80

F601 B9 3C 00                   LDA     INPT4,Y         ;IF BUTTON PRESSED THEN DROP PIECE IF LEGAL
F604 A8                         TAY
F605 45 EA                      EOR     OLDINP
F607 30 1B                      BMI     NOCHDO
F609 98                         TYA                     ;NO CHANGE IN PUSHBUTTON
F60A 30 07                      BMI     JPM05           ;NOT PRESSED
F60C A5 ED                      LDA     ILEGAL          ;BUTTON PRESSED
F60E D0 57                      BNE     JOY20           ;ILLEGAL NOISE SOUND
F610 4C 66 F7           JOY111  JMP     CRSBLK          ;OK-- WAIT FOR LETUP
F613                    JPM05
F613 A5 ED                      LDA     ILEGAL          ;JOYSTICK MOVEMENT ERROR?
F615 10 06                      BPL     JPM06           ;NO.
F617 A5 BB                      LDA     ACTIVE          ;SET UP CARRY TO INDICATE PLAYER
F619 2A                         ROL     A
F61A 4C DD F6                   JMP     MOVE12          ;YES.  CHECK JOYSTICK MOTION
F61D                    JPM06
F61D A9 00                      LDA     #0              ;OK
F61F 85 ED                      STA     ILEGAL
F621                    JPM10
F621 4C D4 F6                   JMP     MOVE10          ;CHECK JOYSTICK
F624                    NOCHDO
F624 84 EA                      STY     OLDINP          ;STORE NEW BUTTON READING
F626 98                         TYA
F627 30 F8                      BMI     JPM10           ;NOT PRESSED

                        ;                               BUTTON JUST PRESSED
F629 A5 E7                      LDA     FROMTO
F62B D0 46                      BNE     TO10            ;TO

                        ;                               FROM
F62D 20 E2 FA                   JSR     TOSET           ;DON'T DISPLAY COMPUTER'S MOVE, SET UP TOSQRB
F630 A6 AB                      LDX     CURSOR
F632 20 2D FA                   JSR     MOVCHK
F635 90 06                      BCC     FROM10          ;CAN JUMP
F637 F0 1F                      BEQ     JOY12           ;CAN'T MOVE

F639 A5 A5                      LDA     MSTJMP          ;MUST JUMP?
F63B D0 1B                      BNE     JOY12           ;YES, NOT LEGAL PIECE TO PICK UP
F63D                    FROM10
F63D E6 E7                      INC     FROMTO          ;CAN MOVE - DO TO NEXT

F63F A6 AB                      LDX     CURSOR
F641 86 E9                      STX     MOVE
F643 B5 B0                      LDA     BOARD,X
F645 85 A3                      STA     PIECE
F647 85 E4                      STA     MOVVAL
F649 A9 00                      LDA     #0
F64B 85 A6                      STA     MLTJMP          ;NO MULTIPLE JUMPS YET
F64D 20 F3 FA                   JSR     TONORM
F650 A9 30                      LDA     #PICKUP
F652 20 EB FF                   JSR     SOUND1
F655 4C D6 F7                   JMP     ATTR10
```

```
                        ;
                        ;       ILLEGAL MOVE -- MAKE NOISE
                        ;
F658                    JOY12
F658 A9 01                      LDA     #1
F65A                    JOY13
F65A 85 ED                      STA     ILEGAL          ;ILLEGAL MOVE NOISE WILL CONTINUE UNTIL BUTTON IS RELEASED
F65C 20 F3 FA                   JSR     TONORM
F65F A5 A5                      LDA     MSTJMP
F661 F0 04                      BEQ     JOY20
F663 A9 BB                      LDA     #JP
F665 85 AD                      STA     TOSQRB
F667                    JOY20
F667 A9 30                      LDA     #$30            ;AUDF0
F669 A2 08                      LDX     #$08            ;AUDV0
F66B A0 0F                      LDY     #ERRSND         ;AUDC0
F66D 20 EF FF                   JSR     SOUND3
F670 4C 66 F7                   JMP     CRSBLK
                        ;
                        ;       "TO" MODE
                        ;
F673                    TO10
F673 A6 AB                      LDX     CURSOR          ;FROM SQUARE
F675 A9 00                      LDA     #0
F677 95 80                      STA     BOARD,X
F679 E4 E9                      CPX     MOVE            ;FROM SQUARE = TO SQUARE?
F67B D0 0E                      BNE     TO11            ;NO.
F67D A4 A6                      LDY     MLTJMP          ;YES.  MULTIPLE JUMP?
F67F D0 D7                      BNE     JOY12           ;YES.   MUST MOVE THIS PIECE TO CONTINUE MULTIPLE JUMP
F681 20 DE FE                   JSR     TOMOV           ;SET UP TOSQRB, CLEAR MOVFLG.  RETURN TO "FROM" SQUARE -- CLEAR MOVE
F684 A5 A3                      LDA     PIECE
F686 95 80                      STA     BOARD,X
F688 4C CA F6                   JMP     JOY81           ;WAIT FOR SAME PLAYER TO MOVE AGAIN
F68B                    TO11
F68B A4 A5                      LDY     MSTJMP          ;MUST JUMP?
F68D F0 29                      BEQ     TO50            ;NO.  SKIP
                        ;
                        ;       JUMP
                        ;
                        ;
                        ;       CHANGE OPPONENT'S PIECE COUNT -- REMOVE JUMPED PIECE
                        ;
F68F 8A                         TXA                     ;CURSOR
F690 A4 BE                      LDY      INDEX
F692 18                         CLC
F693 79 66 FF                   ADC     MOVTAB,Y
F696 AA                         TAX
F697 B5 80                      LDA     BOARD,X         ;LOAD JUMPED PIECE

F699 A0 00                      LDY     #0
F69B 94 80                      STY     BOARD,X         ;REMOVE JUMPED PIECE FROM BOARD
F69D 0A                         ASL     A               ;SHIFT KING BIT TO MSB
F69E 0A                         ASL     A
F69F 20 51 FE                   JSR     CHGCNT          ;CHANGE OPPONENT'S PIECE COUNT
F6A2 20 D1 FF                   JSR     TOSUB           ;DO STUFF WITH PIECE INCLUDING KING CHECK
                        ;                               STORE IN NEW SQUARE, RETURN X=MOVE.  NEW KING?
F6A5 B0 14                      BCS     TO51            ;YES.   NEW KING CAN'T DO MULTIPLE JUMP
F6A7 20 2D FA                   JSR     MOVCHK          ;RETURN X UNCHANGED.   CAN PIECE JUMP AGAIN?
F6AA B0 0F                      BCS     TO51            ;NO.
                        ;                               YES.
```

```
                        .IF     PRNT
                        JSR     PRMOV2          ;PRINT MOVE IF DEBUG
                        .ENDIF

F6AC 86 AB              STX     CURSOR          ;MOVE MOVE TO CURSOR

F6AE A9 BB              LDA     #JP
F6B0 85 AD              STA     TOSQRB
F6B2 85 E7              STA     FROMTO          ;TO MODE
F6B4 85 A6              STA     MLTJMP          ;<>0 => CONTINUE JUMPING
F6B6 D0 16              BNE     JOY90           ;JMP

F6B8            TO50
F6B8 20 D1 FF           JSR     TOSUB           ;DO STUFF WITH MOVE & PIECE INCLUDING KING CHECK & STORE
F6BB            TO51
                        .IF     PRNT
                        JSR     PRMOV2
                        .ENDIF

F6BB 86 AB              STX     CURSOR
F6BD 20 F3 FA           JSR     TONORM          ;RETURNS Y=$FF
F6C0 C8                 INY                     ;0
F6C1 84 A5              STY     MSTJMP          ;CLEAR MSTJMP

F6C3 20 14 FB           JSR     SWIT            ;REVERSE ACTIVE PLAYER
F6C6 A9 2D              LDA     #FRZTIM         ;SET FREEZE TIMER
F6C8 85 F2              STA     FREEZE

                        ;
                        ;       BRANCH HERE FROM SETUP MODE WHEN PUSHBUTTON PUSHED
                        ;       BRANCH HERE FROM CASE WHERE PIECE IS RETURNED TO STARTING SQUARE
                        ;
F6CA            JOY81
F6CA A9 00              LDA     #0
F6CC 85 E7              STA     FROMTO          ;DO FROM NEXT
F6CE            JOY90
F6CE 20 E9 FF           JSR     SOUNDO          ;PIECE DROPPING SOUND
F6D1 4C D6 F7           JMP     ATTR10
```

```
                        ;
F6D4                    MOVE10                          CHECK FOR JOYSTICK MOVEMENT EVERY FEW FRAMES
F6D4 A5 BB                      LDA     ACTIVE          ;MOVE BIT 7 OF ACTIVE TO BIT 1 OF Y
F6D6 2A                         ROL     A               ;SET CARRY IF PLAYER 1
                        ;
                        ;       BRANCH HERE FROM SETUP WITH Y=0   CC => PLAYER 0
                        ;
F6D7                    BNEO
F6D7 A5 C3                      LDA     FRAME
F6D9 29 1F                      AND     #$1F
F6DB D0 27                      BNE     MOVE36          ;DON'T MOVE THIS FRAME
F6DD                    MOVE12
F6DD AD 80 02                   LDA     SWCHA
F6E0 B0 04                      BCS     MOVE20          ;CHECK CARRY BIT (SET UP PREVIOUSLY)
F6E2 4A                         LSR     A               ;P0
F6E3 4A                         LSR     A
F6E4 4A                         LSR     A
F6E5 4A                         LSR     A
F6E6                    MOVE20
F6E6 29 0F                      AND     #$F
F6E8 C9 0F                      CMP     #$F             ;IS JOYSTICK CENTERED?
F6EA F0 18                      BEQ     MOVE36          ;YES.  SKIP
                        ;
                        ;                               ;JOYSTICK IS NOT CENTERED
F6EC                    MOVE28
F6EC 20 F3 FA                   JSR     TONORM

F6EF A6 E9                      LDX     MOVE            ;TO MODE
F6F1 A4 E7                      LDY     FROMTO          ;FROM?
F6F3 D0 05                      BNE     MOVE29          ;NO.  TO

F6F5 20 DE FE                   JSR     TOMOV           ;SET UP TOSQRB, CLEAR MOVFLG.  RETURN TO "FROM" SQUARE -- CLEAR MOVE
F6F8 A6 AB                      LDX     CURSOR
F6FA                    MOVE29

F6FA A0 03                      LDY     #3              ;CHECK FOR DIAGONAL MOVEMENT
F6FC                    M28LP
F6FC D9 62 FF                   CMP     JOYTAB,Y
F6FF F0 06                      BEQ     MOVE37          ;DIAGONAL FOUND
F701 88                         DEY
F702 10 F8                      BPL     M28LP           ;CONTINUE
F704                    MOVE36
F704 4C 62 F7                   JMP     FRM30           ;NOT DIAGONAL -- NO SOUND

                        ;                               ;DIAGONAL MOVEMENT
F707                    MOVE37
F707 20 C9 FA                   JSR     ONBRD2          ;COMPUTE MOVE SQUARE.  OFF BOARD?
F70A 90 05                      BCC     MOVE50          ;NO.  OK
F70C                    MOVE40
F70C A9 80                      LDA     #$80            ;INDICATE JOYSTICK ILLEGAL MOVE
F70E 4C 5A F6                   JMP     JOY13           ;ERROR
F711                    MOVE50
F711 AA                         TAX
F712 A5 E7                      LDA     FROMTO          ;FROM?
F714 D0 09                      BNE     FRM10           ;NO.  TO
F716 B5 80                      LDA     BOARD,X         ;YES.  FROM
F718 85 A3                      STA     PIECE
F71A 86 AB                      STX     CURSOR
F71C 4C 5B F7                   JMP     FRM20           ;JMP
F71F                    FRM10
```

```
F71F A5 A5                      LDA     MSTJMP          ;MUST JUMP?
F721 F0 0E                      BEQ     MOVE60          ;NO.
F723 B5 80                      LDA     BOARD,X         ;YES.  SQUARE EMPTY?
F725 F0 E5                      BEQ     MOVE40          ;YES.  ERROR
F727 45 BB                      EOR     ACTIVE          ;NO.  OPPONENT'S PIECE?
F729 10 E1                      BPL     MOVE40          ;NO.  ERROR
F72B 20 C9 FA                   JSR     ONBRD2          ;YES.  MOVE TO NEXT SQUARE.   OFF BOARD?
F72E B0 DC                      BCS     MOVE40          ;YES.  ERROR
F730 AA                         TAX
F731                    MOVE60
F731 E4 AB                      CPX     CURSOR          ;RETURN TO FROM SQUARE?
F733 F0 1C                      BEQ     FRM15           ;YES.  OK
F735 A5 E9                      LDA     MOVE            ;ALREADY MOVED?
F737 C5 AB                      CMP     CURSOR
F739 D0 D1                      BNE     MOVE40          ;YES.  CAN'T MOVE PIECE AGAIN (EXCEPT TO FROM SQUARE)
F73B B5 80                      LDA     BOARD,X         ;EMPTY?
F73D D0 CD                      BNE     MOVE40          ;NO.  ERROR
F73F A5 A3                      LDA     PIECE           ;YES.  LEGAL MOVE DIRECTION?
F741 C9 10                      CMP     #HUMCHK         ;HUMAN CHECKER (NOT KING)?
F743 D0 04                      BNE     MOVE70          ;NO.
F745 C0 02                      CPY     #2              ;YES.  LEGAL?
F747 90 C3                      BCC     MOVE40          ;NO.  ERROR
F749                    MOVE70
F749 C9 90                      CMP     #CMPCHK         ;RIGHT PLAYER CHECKER (NOT KING)?
F74B D0 04                      BNE     FRM15           ;NO.  SKIP
F74D C0 02                      CPY     #2              ;YES.  LEGAL?
F74F B0 BB                      BCS     MOVE40          ;NO.  ERROR
F751                    FRM15
F751 B5 80                      LDA     BOARD,X
F753 85 E4                      STA     MOVVAL
F755 86 E9                      STX     MOVE
F757 86 AD                      STX     TOSQRB
F759 84 BE                      STY     INDEX

F75B                    FRM20
F75B A9 30                      LDA     #$30            ;SET UP FOR MOVE SOUND  -- LOAD AUDF0
F75D A0 0E                      LDY     #14             ;AUDC0
F75F 20 ED FF                   JSR     SOUND2          ;USE MAX VOLUME -- CALL SOUND ROUTINE
F762                    FRM30
F762 A9 00                      LDA     #0
F764 85 ED                      STA     ILEGAL          ;CLEAR ILLEGAL MOVE FLAG
```

```
                        ;       SET UP CURSOR, MOVE, JMP DISPLAY
F766                    CRSBLK
F766 A5 C3                      LDA     FRAME
F768 29 10                      AND     #$10
F76A D0 27                      BNE     CRSB10
F76C A5 E5                      LDA     MOVFLG
F76E F0 66                      BEQ     ATTR10

F770 38                         SEC                     ;DISPLAY COMPUTER MOVE
F771 20 CE F9                   JSR     JMPSET          ;DISPLAY ANY JUMPED PIECES
F774 A6 E8                      LDX     CURSC           ;COMPUTER CURSOR
F776 A9 B0                      LDA     #CMPCRS
F778 95 80                      STA     BOARD,X
F77A A6 E9                      LDX     MOVE            ;DON'T SHOW COMPUTER'S NEW LOC
F77C A9 00                      LDA     #0
F77E 95 80                      STA     BOARD,X

F780 A6 AB                      LDX     CURSOR
F782 E4 E9                      CPX     MOVE            ;HUMAN CURSOR & COMPUTER MOVE IN SAME SQUARE?
F784 D0 50                      BNE     ATTR10          ;NO
F786 A9 30                      LDA     #HUMCRS         ;YES.  DISPLAY HUMAN CURSOR NOW
F788 A4 BB                      LDY     ACTIVE
F78A 10 02                      BPL     CRSB03
F78C A9 B0                      LDA     #CMPCRS
F78E                    CRSB03
F78E 95 80                      STA     BOARD,X
F790 4C D6 F7                   JMP     ATTR10          ;JMP

F793                    CRSB10
F793 A5 E6                      LDA     WINSAV
F795 F0 0C                      BEQ     CRSB13

F797 A5 E5                      LDA     MOVFLG
F799 F0 22                      BEQ     CRSB17
F79B A6 E8                      LDX     CURSC           ;COMPUTER HAS WON -- DISPLAY ITS CURSOR
F79D A9 B0                      LDA     #CMPCRS
F79F 95 80                      STA     BOARD,X
F7A1 D0 1A                      BNE     CRSB17          ;JMP

F7A3                    CRSB13
F7A3 A6 AB                      LDX     CURSOR          ;HUMAN CURSOR
F7A5 A9 00                      LDA     #0
F7A7 A4 F2                      LDY     FREEZE          ;FREEZE LAST MOVE?
F7A9 D0 10                      BNE     CRSB15          ;YES.  DON'T DISPLAY CURSOR OR PIECE -- PIECE WILL BLINK
F7AB A5 E5                      LDA     MOVFLG          ;COMPUTER MOVE BEING DISPLAYED?
F7AD F0 04                      BEQ     CRSB14          ;NO.  OK
F7AF E4 E9                      CPX     MOVE            ;HUMAN CURSOR & COMPUTER MOVE ON SAME SQUARE?
F7B1 F0 0A                      BEQ     CRSB17          ;YES.  DON'T DISPLAY HUMAN CURSOR NOW
F7B3                    CRSB14
F7B3 A9 30                      LDA     #HUMCRS
F7B5 A4 BB                      LDY     ACTIVE
F7B7 10 02                      BPL     CRSB15
F7B9 A9 B0                      LDA     #CMPCRS         ;RIGHT PLAYER (P1) IS ACTIVE
F7BB                    CRSB15
F7BB 95 80                      STA     BOARD,X

F7BD                    CRSB17
F7BD A5 E7                      LDA     FROMTO
F7BF F0 15                      BEQ     ATTR10          ;FROM (MAY BE DISPLAYING COMPUTER MOVE)
F7C1 A6 E9                      LDX     MOVE            ;TO
```

```
F7C3 A4 A3                  LDY     PIECE
F7C5 C4 E4                  CPY     MOVVAL          ;DOES PIECE = MOVVAL?
F7C7 D0 0B                  BNE     CRSB20          ;NO.  SKIP
F7C9 A5 C3                  LDA     FRAME           ;YES.  ONLY DISPLAY PIECE ON XXX10100 - XXX11011
F7CB 6A                     ROR     A               ;WHEN B2 & B3 ARE DIFFERENT.
F7CC 45 C3                  EOR     FRAME
F7CE 29 04                  AND     #4
F7D0 D0 02                  BNE     CRSB20
F7D2 A0 00                  LDY     #0
F7D4                CRSB20
F7D4 94 80                  STY     BOARD,X
```

```
                        ;       LOAD RAM TABLE OF GRAPHICS FOR SCORE KERNEL
F7D6                    ATTR10
F7D6 A0 01                      LDY     #1              ;ONE PLAYER
F7D8 A6 B0                      LDX     GAMNO
F7DA 86 F6                      STX     SCP0
F7DC E0 10                      CPX     #PLAY2
F7DE D0 01                      BNE     ATTR20
F7E0 C8                         INY                     ;2-PLAYER GAME
F7E1                    ATTR20
F7E1 84 F7                      STY     SCP1

F7E3 A5 F2                      LDA     FREEZE          ;FREEZE?
F7E5 D0 0D                      BNE     ATTR35          ;YES. SKIP -- DON'T CHANGE FROM
F7E7 A6 AB                      LDX     CURSOR
F7E9 A5 E5                      LDA     MOVFLG          ;COMPUTER MOVE DISPLAY?
F7EB F0 02                      BEQ     ATTR30          ;NO.
F7ED A6 AC                      LDX     FROMB           ;YES. USE FROMB
F7EF                    ATTR30
F7EF 20 8D FA                   JSR     IEDSP
F7F2 85 F4                      STA     FROM            ;CONVERT BINARY TO BCD
F7F4                    ATTR35
F7F4 A6 AD                      LDX     TOSQRB
F7F6 20 8D FA                   JSR     IEDSP
F7F9 85 F5                      STA     TOSQR


                        ;
                        ;       BASED ON VCS BASKETBALL BY AL MILLER (WITH EOR IMPROVEMENT A LA COIN-OP)
                        ;
F7FB A2 00                      LDX     #0
F7FD 86 CF                      STX     T4+1            ;INITIALIZE HIGH BYTE INDIRECT POINTER
F7FF A9 FF                      LDA     #SCRTBL/256
F801 85 B4                      STA     T0+1
F803 85 B6                      STA     T2+1
F805 A9 E4                      LDA     #5*4+SCRLPO
                        ;
F807 A2 03                      LDX     #3
F809                    LODRAM
F809 38                         SEC
F80A E9 05                      SBC     #5
F80C 85 CE                      STA     T4
F80E B5 F4                      LDA     FROM,X          ;GET LOW ORDER DIGIT
F810 29 0F                      AND     #$0F
F812 85 B5                      STA     T2
F814 0A                         ASL     A
F815 0A                         ASL     A               ;SHIFT LEFT AND CLEAR CARRY
F816 65 B5                      ADC     T2              ;FIVE TIMES LOW DIGIT
F818 69 8C                      ADC     #SCRTBL&$FF             ;ADD IN BEGINNING OF GRAPHICS TABLE
F81A 85 B3                      STA     T0              ;STORE IN LOW BYTE OF INDIRECT POINTER
                        ;
                        ;
F81C B5 F4                      LDA     FROM,X          ;GET HIGH ORDER DIGIT
F81E 29 F0                      AND     #$F0
F820 4A                         LSR     A
F821 4A                         LSR     A               ;SHIFT RIGHT AND CLEAR CARRY
F822 85 B5                      STA     T2
F824 4A                         LSR     A
F825 4A                         LSR     A
F826 65 B5                      ADC     T2              ;FIVE TIMES HIGH ORDER DIGIT
F828 69 8C                      ADC     #SCRTBL&$FF             ;ADD IN BEGINNING OF GRAPHICS TABLE
```

```
F82A 85 B5                      STA     T2              ;STORE IN LOW BYTE OF INDIRECT POINTER
                        ;
F82C A0 04                      LDY     #4
F82E                    HIDIGT
F82E B1 B3                      LDA     (T0),Y
F830 51 B5                      EOR     (T2),Y          ;TAKE UPPER NIBBLE FROM T2, LOWER FROM T0
F832 29 0F                      AND     #$0F
F834 51 B5                      EOR     (T2),Y
F836 91 CE                      STA     (T4),Y          ;STORE IN RAM TABLE OF GRAPHICS
F838 88                         DEY
F839 10 F3                      BPL     HIDIGT

F83B A5 CE                      LDA     T4
                        ;
F83D CA                         DEX
F83E 10 C9                      BPL     LODRAM
                        ;
F840 A2 05                      LDX     #5
F842 B5 DE              DOROT   LDA     SCRRPF-1,X      MAKE MIRROR IMAGE
                        ;
F844 A0 07                      LDY     #7
F846 2A                 ROTIT   ROL     A
F847 76 DE                      ROR     SCRRPF-1,X
F849 88                         DEY
F84A 10 FA                      BPL     ROTIT
                        ;
F84C CA                         DEX
F84D D0 F3                      BNE     DOROT
```

```
                        ;
                        ;       SET UP PLAYERS AND MISSILE FOR SCORE
                        ;
F84F A9 15              LDA     #$15
F851 85 04              STA     NUSIZ0          ;2 CLOCK MISSILE, DOUBLE SIZE PLAYERS (1 COPY)
F853 85 05              STA     NUSIZ1
F855 A9 F0              LDA     #$F0            ;1 TO RIGHT
F857 85 20              STA     HMP0
F859 A9 30              LDA     #$30            ;3 TO LEFT
F85B 85 21              STA     HMP1
F85D A9 C0              LDA     #$C0            ;4 TO RIGHT
F85F 85 22              STA     HMM0

F861 85 02              STA     WSYNC
F863 A0 08              LDY     #8              ;8*5+1 = 41 MACHINE CYCLES
F865 88         WSLP    DEY                     ;WARNING -- MUST NOT CROSS PAGE BOUNDARY
F866 D0 FD              BNE     WSLP            ;DELAY
F868 85 10              STA     RESP0           ;44
F86A EA                 NOP                     ;46  DELAY
F86B 85 12              STA     RESM0           ;49
F86D 85 11              STA     RESP1           ;52

F86F 85 02              STA     WSYNC
F871 85 2A              STA     HMOVE
```

```
                        ;
                        ;       SET UP COLORS, DOING ATTRACT IF NECESSARY
                        ;
F873 A2 00                      LDX     #0              ;0 IF NOT ATTRACT
F875 A5 C2                      LDA     OLDPBQ
F877 29 08                      AND     #$08            ;B/W?
F879 D0 3E                      BNE     QQ30            ;NO. COLOR -- SKIP
F87B A9 0F                      LDA     #$0F            ;YES.
F87D A4 EF                      LDY     GFLG2           ;ATTRACT?
F87F D0 04                      BNE     QQ10            ;NO. SKIP
F881 A6 EE                      LDX     ATIM            ;YES. USE ATTRACT TIMER TO CHANGE LUMS
F883 A9 07                      LDA     #$07            ;MASK OFF COLOR BITS
F885                    QQ10
F885 85 B3                      STA     T0              ;SAVE MASK
F887 8A                         TXA
F888 49 06                      EOR     #$06            ;DARK SQUARE COLOR
F88A 25 B3                      AND     T0
F88C 85 F3                      STA     COLSQ
F88E 85 08                      STA     COLUPF
F890 85 09                      STA     COLUBK          ;USE THIS FOR BK COLOR FOR SCORE
F892 8A                         TXA
F893 49 0A                      EOR     #$0A            ;LIGHT SQUARE COLOR
F895 25 B3                      AND     T0
F897 85 BA                      STA     T8
F899 8A                         TXA
F89A 49 0C                      EOR     #$0C            ;LIGHT CHECKER COLOR
F89C 25 B3                      AND     T0
F89E 85 CF                      STA     COLP1           ;ASSUME COMPUTER (P1) IS LIGHT
F8A0 8A                         TXA
                        ;       EOR     #$00            ;ASSUME HUMAN (P0) IS DARK
F8A1 25 B3                      AND     T0
F8A3 85 CE                      STA     COLP0
F8A5 A4 B1                      LDY     COLHUM
F8A7 C0 36                      CPY     #DARK           ;IS HUMAN DARK?
F8A9 F0 07                      BEQ     QQ20            ;YES. SKIP
F8AB A4 CF                      LDY     COLP1           ;NO. SWITCH
F8AD 84 CE                      STY     COLP0
F8AF 85 CF                      STA     COLP1
F8B1 98                         TYA
F8B2                    QQ20
F8B2 85 06                      STA     COLUP0          ;STORE HUMAN COLOR FOR SCORE
F8B4 85 07                      STA     COLUP1
F8B6 4C 0F F9                   JMP     QQ50
                        ;
                        ;       COLOR
                        ;
F8B9                    QQ30
F8B9 A9 FF                      LDA     #$FF
F8BB A4 EF                      LDY     GFLG2           ;ATTRACT?
F8BD D0 09                      BNE     QQ40            ;NO. SKIP
F8BF A5 EE                      LDA     ATIM            ;YES. ROTATE SO COLORS SHIFT QUICKLY
F8C1 2A                         ROL     A
F8C2 2A                         ROL     A
F8C3 2A                         ROL     A
F8C4 2A                         ROL     A
F8C5 AA                         TAX
F8C6 A9 F7                      LDA     #$F7            ;MASK OFF HI BIT OF LUM IN ATTRACT
F8C8                    QQ40
F8C8 85 B3                      STA     T0
F8CA 8A                         TXA
```

```
F8CB 49 92                  EOR     #DRKCHR
F8CD A4 B1                  LDY     COLHUM
F8CF C0 36                  CPY     #DARK           ;IS HUMAN DARK?
F8D1 F0 03                  BEQ     LPSKIP          ;YES.  SKIP
F8D3 8A                     TXA
F8D4 49 0C                  EOR     #LTCHR          ;HUMAN IS LIGHT -- SO IS SCORE COLOR
F8D6                LPSKIP
F8D6 25 B3                  AND     TO
F8D8 85 06                  STA     COLUPO
F8DA 85 07                  STA     COLUP1

F8DC A0 00                  LDY     #0
F8DE A5 B0                  LDA     GAMNO
F8E0 29 0F                  AND     #$0F
F8E2 C9 05                  CMP     #HMIN
F8E4 90 02                  BCC     LPX
F8E6 A0 03                  LDY     #3              ;USE DIFFERENT COLORS FOR HARD GAMES
F8E8                LPX
F8E8 8A                     TXA
F8E9 59 CB FF               EOR     COLTB,Y
F8EC 25 B3                  AND     TO
F8EE 85 F3                  STA     COLSQ           ;DARK SQUARE COLOR
F8F0 85 08                  STA     COLUPF          ;SET UP PLAYFIELD COLOR

F8F2 A5 EF                  LDA     GFLG2           ;ATTRACT?
F8F4 D0 01                  BNE     LPQX            ;NO.
F8F6 C8                     INY                     ;YES.  USE DIF COLORS
F8F7                LPQX
F8F7 8A                     TXA
F8F8 59 CC FF               EOR     COLTB+1,Y
F8FB 25 B3                  AND     TO
F8FD 85 09                  STA     COLUBK          ;LIGHT SQUARE COLOR AND BK
F8FF 85 BA                  STA     T8

F901 8A                     TXA
F902 45 B1                  EOR     COLHUM
F904 25 B3                  AND     TO
F906 85 CE                  STA     COLPO
F908 8A                     TXA
F909 45 B2                  EOR     COLCMP
F90B 25 B3                  AND     TO
F90D 85 CF                  STA     COLP1

F90F                QQ50
F90F A5 E6                  LDA     WINSAV
F911 F0 12                  BEQ     LP10
F913 A5 C3                  LDA     FRAME           ;SOMEONE HAS WON
F915 29 10                  AND     #$10
F917 D0 0C                  BNE     LP10            ;TURN OFF REMAINING PLAYER'S PIECES EVERY
                    ;                               FEW FRAMES BY SETTING TO PF COLOR.
F919 A5 F3                  LDA     COLSQ
F91B A6 BB                  LDX     ACTIVE
F91D 30 04                  BMI     LP06
F91F 85 CF                  STA     COLP1           ;HUMAN CAN'T MOVE -- TURN OFF COMPUTER
F921 10 02                  BPL     LP10            ;JMP
F923                LP06
F923 85 CE                  STA     COLPO           ;COMPUTER CAN'T MOVE -- TURN OFF HUMAN
F925                LP10
```

VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM

```
F925 A4 F0                  LDY     BLNKTM
F927 F0 0F                  BEQ     VBLP

F929 A0 02                  LDY     #2              ;KEEP VBLANK ON WHILE SYNCING AFTER CALCULATIONS

F92B C6 F0                  DEC     BLNKTM
F92D D0 09                  BNE     VBLP

F92F A5 B0                  LDA     GAMNO           ;BLANK TIME IS UP
F931 C9 10                  CMP     #PLAY2
F933 F0 03                  BEQ     VBLP            ;2-PLAYER GAME
F935 20 E9 FF               JSR     SOUNDO          ;PIECE DROPPING SOUND IF 1-PLAY & HUMAN NEXT

F938 AE 84 02       VBLP    LDX     INTIM           ;WAIT FOR TIMER TO RUN OUT
F93B D0 FB                  BNE     VBLP

F93D 85 02                  STA     WSYNC
F93F 84 01                  STY     VBLANK          ;END OF VERTICAL BLANK (IF Y=0)
                            .IF     PAL             ;IF PAL THEN ADD 3 EXTRA LINES AT TOP OF DISPLAY
                            JSR     WAIT3           ;DO 3 WSYNC'S
                            .ENDIF

F941 4C 00 F0               JMP     JMPBAK          ;SKIP BACK TO KERNEL
;                                   END OF MAIN PROGRAM
```

```
                        ;       GSRST -- SUBROUTINE TO CHECK GAME SELECT AND RESET SWITCHES
                        ;
F944                    GSRST
F944 A5 C2                      LDA     OLDPBQ          ;CHECK GAME SELECT
F946 6A                         ROR     A
F947 6A                         ROR     A
F948 B0 23                      BCS     GS1             ;BRANCH IF GAME SELECT NOT PRESSED

F94A C6 EB                      DEC     GSTIM           ;GAME SELECT PRESSED -- DON'T CHANGE GAME NUMBER TILL TIME UP
F94C D0 1E                      BNE     GRTN2

F94E A9 2D                      LDA     #GSCON
F950 85 EB                      STA     GSTIM           ;REINITIALIZE TIMER

F952 F8                         SED                     ;DECIMAL ARITHMETIC
F953 A5 B0                      LDA     GAMNO
F955 18                         CLC
F956 69 01                      ADC     #1
F958 C9 20                      CMP     #$20            ;GMAX+1 IN DECIMAL
F95A D0 02                      BNE     RESET2
F95C A9 01                      LDA     #1
F95E                    RESET2
F95E 85 B0                      STA     GAMNO
F960 D8                         CLD

F961 A5 E6                      LDA     WINSAV          ;STAY IN ATTRACT IF WIN
F963 D0 03                      BNE     GSKIP

F965 20 F3 FA                   JSR     TONORM          ;MAKE SURE NOT ATTRACT, RETURN Y=$FF
F968                    GSKIP
F968 A0 00                      LDY     #0
F96A 84 1A                      STY     AUDV1
F96C                    GRTN2
F96C 60                         RTS

                        ;                               ;SET UP GAME SELECT TIMER & CHECK GAME RESET
F96D                    GS1
F96D 0A                         ASL     A
F96E B0 59                      BCS     RES20           ;NOT PRESSED -- GSTIM<-1 & RETURN

F970 A9 00                      LDA     #0              ;RESET BUTTON IS PRESSED
F972 A2 2F                      LDX     #ZROEND-1-ZROBGN
F974                    CLRLP
F974 95 80                      STA     ZROBGN,X
F976 CA                         DEX
F977 10 FB                      BPL     CLRLP

                                .IF     PRNT
                                LDX     #0
                                STX     PRCNT
                                .ENDIF

F979 85 1A                      STA     AUDV1           ;OFF (0)

F97B A2 03                      LDX     #3              ;CLEAR MORE STUFF -- MOVVAL, MOVFLG, WINSAV, FROMTO
F97D                    RESLP
F97D 95 E4                      STA     MOVVAL,X
F97F CA                         DEX
```

```
F97F CA                 DEX

F980 10 FB              BPL     RESLP

                ;
                ;       RESET -- ENTRY POINT FROM INIT CODE
                ;
F982            RESET
F982 20 F3 FA           JSR     TONORM          ;NOT ATTRACT, Y<-$FF
F985 84 A4              STY     JMPI            ;$FF

F987 20 E1 FE           JSR     MOVCLR          ;NO CURRENT MOVE SO POINT OFF BOARD
F98A A5 EC              LDA     OLDPB6
F98C D0 17              BNE     RES10           ;IF SETUP MODE THEN LEAVE BOARD EMPTY
                ;                               SET UP BOARD
F98E A9 10              LDA     #CHECKER+CLP0   ;HUMAN ON 0-12 (8 NOT USED)
F990 A0 90              LDY     #CHECKER+CLP1   ;COMPUTER ON SQUARES 22-34 (26 NOT USED)
F992 A2 0D              LDX     #13-1+1
F994            BRDLP
F994 95 7F              STA     BOARD-1,X
F996 94 95              STY     BOARD-1+22,X
F998 CA                 DEX
F999 D0 F9              BNE     BRDLP

F99B 86 88              STX     BOARD+8         ;CLEAR PHANTOM SQUARES
F99D 86 9A              STX     BOARD+26

F99F A9 18              LDA     #12*2           ;INITIALIZE PIECE COUNTS -- 12 MEN PER SIDE
F9A1 85 AA              STA     CPIECE          ;COMPUTER PIECE COUNT (P1)
F9A3 85 A9              STA     HPIECE          ;HUMAN PIECE COUNT (P0)

F9A5            RES10
F9A5 A2 0B              LDX     #11             ;INITIALIZE CURSOR POSITIONS
F9A7 A0 17              LDY     #23
F9A9 A5 B0              LDA     GAMNO
F9AB C9 10              CMP     #PLAY2
F9AD D0 08              BNE     RES15
F9AF A5 BB              LDA     ACTIVE          ;BOTTOM (P0) PLAYER STARTS?
F9B1 10 04              BPL     RES15           ;YES.
F9B3 A2 17              LDX     #23             ;NO.  TOP PLAYER STARTS
F9B5 A0 0B              LDY     #11
F9B7            RES15
F9B7 86 AB              STX     CURSOR
F9B9 84 E8              STY     CURSC
F9BB B5 80              LDA     BOARD,X
F9BD 85 A3              STA     PIECE

F9BF A9 80              LDA     #$80
F9C1 85 EA              STA     OLDINP
F9C3 20 FB FA           JSR     CRSTUP          ;SET UP ACTIVE, COLHUM, COLCMP BASED ON RIGHT DIFFICULTY SWITCH

F9C6 20 E2 FA           JSR     TOSET           ;MSTJMP=0 SO CLEAR TOSGRB, MOVFLG
F9C9            RES20
F9C9 A2 01              LDX     #1
F9CB 86 EB              STX     GSTIM
F9CD 60                 RTS
```

```
                        .IF     PAL
                ;
                ;       WAIT3 -- DO 3 WSYNC'S
                ;       WAIT2 -- DO 2 WSYNC'S
                ;
                WAIT3
                        STA     WSYNC
                WAIT2
                        STA     WSYNC
                        STA     WSYNC
                        RTS
                        .ENDIF
                ;
                ;       JMPSET -- SET UP OR CLEAR BOARD FOR JUMP DISPLAY
                ;
                ;                       INPUT:  CC  => CLEAR EACH SQUARE
                ;                               CS => PUT JUMPED CHECKERS IN EACH SQUARE IN JMPLST
                ;
F9CE            JMPSET
F9CE A4 A4              LDY     JMPI
F9D0 30 15              BMI     JMPSR           ;NO JUMPS
F9D2            JMPSLP
F9D2 A9 10              LDA     #HUMCHK
F9D4 B6 C5              LDX     JMPLST,Y
F9D6 10 06              BPL     JMPS10
F9D8 8A                 TXA                     ;<0 => KING
F9D9 29 7F              AND     #$7F
F9DB AA                 TAX
F9DC A9 20              LDA     #HUMKNG
F9DE            JMPS10
F9DE B0 02              BCS     JMPS20
F9E0 A9 00              LDA     #0              ;CLEAR SQUARE
F9E2            JMPS20
F9E2 95 80              STA     BOARD,X
F9E4 88                 DEY
F9E5 10 EB              BPL     JMPSLP
F9E7            JMPSR
F9E7 60                 RTS


                ;
                ;       JMPWIN -- SET UP OLDINP, MLTJMP, CALL JMPCHK TO SET UP MSTJMP,XMOVE
                ;                               RETURN EQ=>WIN, NE=> NO WIN
                ;
F9E8            JMPWIN
F9E8 A9 80              LDA     #$80
F9EA 85 EA              STA     OLDINP          ;ASSUME BUTTON NOT PRESSED
F9EC A9 00              LDA     #0
F9EE 85 A6              STA     MLTJMP
F9F0 20 FE F9           JSR     JMPCHK          ;CHECK FOR JUMPS & ANY MOVES
F9F3 D0 08              BNE     JMPRTN          ;CAN MOVE

                        .IF     PRNT            ;CAN'T MOVE -- OTHER PLAYER WINS
                        LDA     PRFLG
                        BEQ     PRSKIP
                        JSR     CRLF
                        LDA     #GAMOVR         ;DISPLAY "GAME OVER" ON TI IF DEBUG
                        JSR     OUTMSG
                        JSR     PRBRD
                PRSKIP
```

VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM

```
                                .ENDIF

F9F5 E6 E6                      INC     WINSAV
F9F7 A9 80                      LDA     #$80
F9F9 85 F1                      STA     SNDTIM
F9FB A9 00                      LDA     #0              ;EQ
F9FD                    JMPRTN
F9FD 60                         RTS
                        ;
                        ;       JMPCHK -- CHECK TO SEE IF ACTIVE PLAYER HAS ANY JUMPS. SETUP MSTJMP, CALL MOVCHK TO SET UP XMOVE
                        ;                                       RETURN CC=> JUMP, CS => NO JUMPS
                        ;                                       MSTJMP=4 OR MSTJMP =0
                        ;                                       NE => ACTIVE PLAYER CAN MOVE (ANYMVE = 1)
                        ;                                       EQ => ACTIVE PLAYER CANNOT MOVE AT ALL (ANYMVE = 0)
                        ;
                        ;                                       SETS UP BLNKTM, TURNS ON VBLANK
                        ;
F9FE                    JMPCHK
F9FE A9 1E                      LDA     #VBTIM          ;BLANK SCREEN FOR 1/2 SECOND AFTER
FA00 85 F0                      STA     BLNKTM          ;THIS BECAUSE IT TAKES 1203-2080 MICROSEC.
FA02 A9 02                      LDA     #2
FA04 85 01                      STA     VBLANK          ;TURN ON VBLANK
                        ;
                        ;       JMPCK2 -- SAME AS JMPCHK EXCEPT NO VBLANK JUNK
                        ;
FA06                    JMPCK2
FA06 A9 00                      LDA     #0
FA08 85 B6                      STA     ANYMVE          ;NO MOVE FOUND YET
FA0A A9 04                      LDA     #4
FA0C 85 A5                      STA     MSTJMP          ;ASSUME JUMP
FA0E A2 22                      LDX     #34
FA10                    JMPLP1
FA10 20 2D FA                   JSR     MOVCHK
FA13 F0 04                      BEQ     JPF1
FA15 E6 B6                      INC     ANYMVE          ;INDICATE THAT MOVE HAS BEEN FOUND
FA17 90 06                      BCC     JMPF2           ;JUMP FOUND
FA19                    JPF1
FA19 CA                         DEX
FA1A 10 F4                      BPL     JMPLP1
FA1C E8                         INX                     ;NO JUMP FOUND  0, CS
FA1D 86 A5                      STX     MSTJMP
FA1F                    JMPF2
FA1F A4 B6                      LDY     ANYMVE
FA21 60                         RTS
```

```
                        ;
                        ;       LOADY -- Y <-- 3 (OR 1 IF CMPCHK)
                        ;
FA22                    LOADY
FA22 A0 03                      LDY     #3
FA24 B5 80                      LDA     BOARD,X
FA26 C9 90                      CMP     #CMPCHK
FA28 D0 02                      BNE     LODY10
FA2A A0 01                      LDY     #1              ;COMPUTER CHECKER CANNOT MOVE FORWARD (UP) -- SKIP RF,LF
FA2C                    LODY10
FA2C 60                         RTS
                        ;
                        ;       MOVCHK -- CHECK TO SEE IF THIS IS AN ACTIVE PIECE WHICH CAN JUMP
                        ;
                        ;                               INPUT: X = SQUARE NUMBER (0-34)
                        ;                               OUTPUT: X=SQUARE #
                        ;                               CC => ACTIVE, CAN JUMP
                        ;                               CS => CANNOT JUMP
                        ;                               Y>0 (NE)  => ACTIVE PIECE WHICH CAN MOVE -- SAVED IN YSAVE
                        ;                               Y=0 (EQ)  => NO MOVE AT ALL FOR THIS SQUARE
                        ;                               XMOVE=START SQUARE OF FIRST MOVE OR JUMP FOUND IF ANYMVE=0 OR IF JUMP.
                        ;
FA2D                    MOVCHK
FA2D A0 00                      LDY     #0
FA2F 84 B5                      STY     YSAVE           ;T2
FA31 B5 80                      LDA     BOARD,X
FA33 F0 44                      BEQ     JMP12           ;EMPTY SQUARE
FA35 45 BB                      EOR     ACTIVE
FA37 30 40                      BMI     JMP12           ;PASSIVE PIECE -- CAN'T MOVE
FA39 86 B4                      STX     XSAVE           ;T1
FA3B 20 22 FA                   JSR     LOADY           ;Y <-- 3 (OR 1 IF CMPCHK)
FA3E                    JMPLP2
FA3E 20 C9 FA                   JSR     ONBRD2          ;ADD MOVTAB OFFSET TO SQUARE AND CHECK TO SEE IF OFF BOARD
FA41 B0 27                      BCS     JMP10           ;OFF BOARD
FA43 AA                         TAX
FA44 B5 80                      LDA     BOARD,X
FA46 D0 0D                      BNE     JMP05
                        ;
                        ;       NOT JUMP
                        ;
FA48 8A                         TXA                     ;NEXT SQUARE IS EMPTY
FA49 A5 B6                      LDA     ANYMVE
FA4B D0 04                      BNE     JMP02
FA4D A6 B4                      LDX     XSAVE
FA4F 86 C0                      STX     XMOVE           ;SAVE START SQUARE AND OFFSET INDEX
FA51                    JMP02
FA51 E6 B5                      INC     YSAVE
FA53 D0 15                      BNE     JMP10           ;JMP
                        ;
                        ;       JUMP
                        ;
FA55                    JMP05
FA55 45 BB                      EOR     ACTIVE          ;NEXT SQUARE IS OCCUPIED
FA57 10 11                      BPL     JMP10           ;ACTIVE PIECE -- CAN'T JUMP OWN PIECE
FA59 20 C9 FA                   JSR     ONBRD2          ;ADD MOVTAB OFFSET TO SQUARE AND CHECK TO SEE IF OFF BOARD
FA5C B0 0C                      BCS     JMP10           ;CAN'T JUMP OFF BOARD
FA5E AA                         TAX
FA5F B5 80                      LDA     BOARD,X
FA61 D0 07                      BNE     JMP10
```

```
FA63 A6 B4                  LDX     XSAVE           ;SAVE START SQUARE AND OFFSET INDEX FOR JUMP
FA65 86 C0                  STX     XMOVE           ;CC, NE  JUMP FOUND
FA67 A0 01                  LDY     #1
FA69 60                     RTS

FA6A                JMP10
FA6A A6 B4                  LDX     XSAVE
FA6C 88                     DEY
FA6D 30 0A                  BMI     JMP12           ;DONE
FA6F C0 01                  CPY     #1
FA71 D0 CB                  BNE     JMPLP2          ;CONTINUE
FA73 B5 80                  LDA     BOARD,X
FA75 C9 10                  CMP     #HUMCHK
FA77 D0 C5                  BNE     JMPLP2
                    ;                               HUMAN CHECKER CAN'T GO BACK (DOWN) SKIP RB,LB
FA79                JMP12
FA79 38                     SEC                     ;NO JUMP FOUND
FA7A A4 B5                  LDY     YSAVE           ;LOAD MOVE COUNT
FA7C 60                     RTS
```

```
                        ;
                        ;       XIE1 --                 INPUT X = INTERNAL SQUARE NUMBER (0-34)
                        ;                               OUTPUT X = EXTERNAL SQUARE #-1    (0-31)
                        ;                               AFFECTS X, CARRY SET IF RESULT = 24-31
                        ;                               INTERNAL #'S 8, 17 AND 26 ARE NOT USED.
                        ;
FA7D                    XIE1
FA7D E0 08                      CPX     #8
FA7F 90 0B                      BCC     ARTN            ;0-7 OK
FA81 CA                         DEX                     ;9-34 TO 8-33
FA82 E0 10                      CPX     #16
FA84 90 06                      BCC     ARTN            ;8-15 OK
FA86 CA                         DEX                     ;17-33 TO 16-32
FA87 E0 18                      CPX     #24
FA89 90 01                      BCC     ARTN            ;16-23 OK
FA8B CA                         DEX                     ;25-32 TO 24-31
FA8C                    ARTN
FA8C 60                         RTS
                        ;
                        ;       IEDSP -- CONVERT INTERNAL SQUARE # IN X (0-34) TO
                        ;                               EXTERNAL IN A FOR DISPLAY.
                        ;                               1-32 IF HUMAN IS DARK, 32-1 IF HUMAN IS LIGHT
FA8D                    IEDSP
FA8D 8A                         TXA
FA8E 10 0A                      BPL     IED04

FA90 A5 C3                      LDA     FRAME
FA92 29 10                      AND     #$10
FA94 F0 02                      BEQ     IED02
FA96 A2 AA                      LDX     #BLANK          ;IF SPECIAL CHAR (E.G. "JP") THEN BLINK ON & OFF
FA98                    IED02
FA98 8A                         TXA
FA99 60                         RTS

FA9A                    IED04
FA9A 20 7D FA                   JSR     XIE1

FA9D A5 C2                      LDA     OLDPBQ          ;CHECK "B/W SWITCH"
FA9F A4 B1                      LDY     COLHUM
FAA1 C0 36                      CPY     #DARK
FAA3 D0 06                      BNE     IED05           ;HUMAN LIGHT => REVERSE NUMBERING
FAA5 29 08                      AND     #$8             ;DARK=>DON'T REVERSE
FAA7 F0 06                      BEQ     IED07           ;B/W => REVERSE
FAA9 D0 0B                      BNE     IED10           ;NOT B/W => DON'T REVERSE
FAAB                    IED05
FAAB 29 08                      AND     #8              ;CHECK B/W
FAAD F0 07                      BEQ     IED10           ;B/W IMPLIES DON'T REVERSE
FAAF                    IED07
FAAF 8A                         TXA                     ;BOARD REVERSED
FAB0 38                         SEC
FAB1 E9 20                      SBC     #32             ;31-X = -(X-31) = -((X-32)+1) = COMP (X-32)
FAB3 49 FF                      EOR     #$FF            ;COMPLEMENT
FAB5 AA                         TAX
FAB6                    IED10
FAB6 E8                         INX                     ;1-32
FAB7 8A                         TXA

                        ;                               CONVERT BINARY IN A TO DECIMAL IN A
FAB8 A0 00                      LDY     #0
FABA                    IEDLP1
```

```
FABA C9 0A                      CMP     #10
FABC 90 07                      BCC     IED20           ;0-9 => DONE
FABE E9 0A                      SBC     #10
FAC0 C8                         INY
FAC1 D0 F7                      BNE     IEDLP1          ;JMP

FAC3                    IEDLP2
FAC3 69 10                      ADC     #16             ;CONVERT BINARY TO DECIMAL
FAC5                    IED20
FAC5 88                         DEY
FAC6 10 FB                      BPL     IEDLP2
FAC8 60                         RTS
```

```
                        ;
                        ;       ONBRD2 -- ADD MOVTAB OFFSET TO SQUARE AND CHECK TO SEE IF OFF BOARD
                        ;
FAC9                    ONBRD2
FAC9 8A                         TXA
FACA 18                         CLC
FACB 79 66 FF                   ADC     MOVTAB,Y
                        ;
                        ;       ONBRD -- INPUT:  A=INTERNAL SQUARE #
                        ;                                       OUTPUT:  CS => OFF BOARD
                        ;                                       A,X,Y ARE NOT CHANGED
                        ;
FACE C9 23                      CMP     #35
FAD0 B0 0E                      BCS     NOTON           ;>=35 OR <0
FAD2 C9 1A                      CMP     #26
FAD4 F0 0A                      BEQ     NOTON
FAD6 C9 11                      CMP     #17
FAD8 F0 06                      BEQ     NOTON
FADA C9 08                      CMP     #8
FADC F0 02                      BEQ     NOTON
FADE 18                         CLC
FADF 60                         RTS
FAE0                    NOTON
FAE0 38                         SEC
FAE1 60                         RTS
```

```
                        ;
                        ;       TOSET -- SET UP TOSQRB, MOVFLG, RETURN Y=0, A,X UNCHANGED, EQ
                        ;
FAE2                    TOSET
FAE2 A4 A5                      LDY     MSTJMP          ;"FROM"
FAE4 F0 04                      BEQ     MV285
FAE6 A0 BB                      LDY     #JP             ;MUST JUMP
FAE8 D0 02                      BNE     MV287           ;JMP
FAEA                    MV285
FAEA A0 AA                      LDY     #BLANK          ;NO JUMP
FAEC                    MV287
FAEC 84 AD                      STY     TOSQRB

FAEE A0 00                      LDY     #0
FAF0 84 E5                      STY     MOVFLG
FAF2 60                         RTS
```

```
                        ;
                        ;       TONORM --       GO OUT OF ATTRACT INTO NORMAL MODE
                        ;
FAF3                    TONORM
FAF3 A0 00                      LDY     #0              ;CLEAR TIMER
FAF5 84 EE                      STY     ATIM
FAF7 88                         DEY                     ;RETURN Y=$FF
FAF8 84 EF                      STY     GFLG2
FAFA 60                         RTS


                        ;
                        ;       CRSTUP -- SET UP ACTIVE AND COLORS (COLHUM&COLCMP) (DEPENDS ON RIGHT DIFFICULTY SWITCH)
                        ;
FAFB                    CRSTUP
FAFB A4 C2                      LDY     OLDPBQ
FAFD 10 08                      BPL     FSET20          ;0 (AM,B,DOWN) => P0 FIRST (ACTIVE)

FAFF A9 80                      LDA     #$80            ;COMPUTER STARTS 1 (PRO,A,UP)
FB01 A2 0C                      LDX     #LIGHT          ;HUMAN IS LIGHT
FB03 A0 36                      LDY     #DARK           ;COMPUTER IS DARK (OR RIGHT HAND PLAYER)
FB05 D0 06                      BNE     FSET30          ;JMP
FB07                    FSET20
FB07 A9 00                      LDA     #0
FB09 A2 36                      LDX     #DARK           ;HUMAN IS DARK
FB0B A0 0C                      LDY     #LIGHT          ;COMPUTER IS LIGHT (OR RIGHT HAND PLAYER)
FB0D                    FSET30
FB0D 85 BB                      STA     ACTIVE
FB0F 86 B1                      STX     COLHUM
FB11 84 B2                      STY     COLCMP
FB13 60                         RTS


                        ;
                        ;       SWIT -- SWITCH ACTIVE PLAYER, RETURN A=ACTIVE
                        ;
FB14                    SWIT
FB14 A5 BB                      LDA     ACTIVE
FB16 49 80                      EOR     #$80
FB18 85 BB                      STA     ACTIVE
FB1A 60                         RTS
```

```
                        ;
                        ;       RESTR6 -- RESTORE A BUNCH OF STUFF INCLUDING FRSMOV
                        ;
FB1B                    RESTR6
FB1B 29 C0                      AND     #$C0
FB1D 85 BF                      STA     FRSMOV
                        ;
                        ;       RESTR4 -- RESTORE BOARD AFTER JUMP
                        ;
FB1F                    RESTR4
FB1F A6 BC                      LDX     DEPTH
FB21 B5 D2                      LDA     OTHER-1,X       ;COMPUTE PIECE
FB23 29 30                      AND     #$30
FB25 05 BB                      ORA     ACTIVE
FB27 85 A3                      STA     PIECE

FB29 B5 D2                      LDA     OTHER-1,X
FB2B 48                         PHA                     ;CAPTURED PIECCE
FB2C A5 AC                      LDA     FROMB           ;FROM SQUARE
FB2E A4 BE                      LDY     INDEX
FB30 18                         CLC
FB31 79 66 FF                   ADC     MOVTAB,Y
FB34 AA                         TAX                     ;JUMP SQUARE
FB35 18                         CLC
FB36 79 66 FF                   ADC     MOVTAB,Y
FB39 85 AD                      STA     TOSQRB
FB3B 68                         PLA
                        ;
                        ;       RESTR5 -- ENTRY POINT
                        ;
FB3C                    RESTR5
FB3C A0 02                      LDY     #2              ;SET UP TO INCREMENT OPPONENT'S PIECE COUNT
FB3E 2A                         ROL     A               ;KING?
FB3F A9 90                      LDA     #CMPCHK
FB41 90 03                      BCC     NOK             ;NO.
FB43 C8                         INY                     ;KING => 3
FB44 A9 A0                      LDA     #CMPKNG
FB46                    NOK
FB46 45 BB                      EOR     ACTIVE          ;OPPOSITE PLAYER -- JUMPED PIECE
FB48 95 80                      STA     BOARD,X

FB4A 29 80                      AND     #$80            ;ACTIVE EOR #$80
FB4C 0A                         ASL     A
FB4D 2A                         ROL     A               ;SHIFT AND CLEAR CARRY
FB4E AA                         TAX
FB4F 98                         TYA
FB50 75 A9                      ADC     HPIECE,X
FB52 95 A9                      STA     HPIECE,X        ;STORE NEW PIECE COUNT

FB54 C0 03                      CPY     #3
FB56 D0 02                      BNE     NOK10
FB58 F6 AE                      INC     HKING,X
FB5A                    NOK10

FB5A A5 A7                      LDA     MSTFLG
FB5C 85 A5                      STA     MSTJMP
FB5E A5 A8                      LDA     MLTFLG          ;RESTORE FLAGS
FB60 85 A6                      STA     MLTJMP
                        ;
```

```
FB60 85 A6                      STA     MLTJMP
                        ;
```



```
                        ;       RESTOR -- MOVE PIECE BACK & DEC PIECE COUNT IF NEW KING
                        ;
FB62                    RESTOR
FB62 A6 AD                      LDX     TOSQRB
FB64 B5 80                      LDA     BOARD,X
FB66 C5 A3                      CMP     PIECE
FB68 F0 09                      BEQ     TERR            ;SAME -- NOT NEW KING
FB6A A5 BB                      LDA     ACTIVE          ;MOVE BIT7 TO BIT 1
FB6C 0A                         ASL     A
FB6D 2A                         ROL     A
FB6E AA                         TAX
FB6F D6 A9                      DEC     HPIECE,X
FB71 D6 AE                      DEC     HKING,X         ;DECREMENT KING COUNT
FB73                    TERR
FB73 A6 AD                      LDX     TOSQRB
FB75 A9 00                      LDA     #0
FB77 95 80                      STA     BOARD,X         ;CLEAR TO SQUARE
FB79 A6 AC                      LDX     FROMB
FB7B A5 A3                      LDA     PIECE           ;MOVE PIECE TO FROM SQUARE
FB7D 95 80                      STA     BOARD,X
FB7F 60                         RTS
```

```
                        ;
                        ;       JUMP1 -- SETS UP JMPLST IF MULTIPLE JUMP FOR COMPUTER (FOR END OF ALPHA-BETA)
                        ;
                        ;       INPUT:  TO = STARTING SQUARE
                        ;               Y = INDEX INTO MOVTAB
                        ;               T1 = CURRENT JMPLST INDEX
                        ;
                        ;       OUTPUT: TO = ENDING SQUARE (AFTER JUMP)
                        ;               T1 = T1+1 (OLD T1 IN X)
                        ;               JMPLST,X = JUMPED PIECE (B7=1 => KING) AND SQUARE
                        ;               CS = DONE (LAST JUMP)
                        ;
FB80                    JUMP1
FB80 A5 B3                      LDA     TO
FB82 18                         CLC
FB83 79 66 FF                   ADC     MOVTAB,Y
FB86 48                         PHA
FB87 AA                         TAX
FB88 B5 80                      LDA     BOARD,X         ;SAVE JUMP SQUARE

FB8A C6 A9                      DEC     HPIECE          ;DECREMENT HUMAN'S PIECE COUNT
FB8C C6 A9                      DEC     HPIECE
FB8E C9 20                      CMP     #HUMKNG
FB90 D0 04                      BNE     NK
FB92 C6 A9                      DEC     HPIECE          ;KING -- DECREMENT AGAIN
FB94 C6 AE                      DEC     HKING           ;DEC KING COUNT
FB96                    NK
FB96 48                         PHA                     ;SAVE JUMPED PIECE
FB97 A9 00                      LDA     #0              ;CLEAR OUT JUMP SQUARE
FB99 95 80                      STA     BOARD,X
FB9B 68                         PLA

FB9C AA                         TAX
FB9D 68                         PLA                     ;JUMP SQUARE
FB9E 48                         PHA
FB9F E0 20                      CPX     #HUMKNG         ;DOES JUMP SQUARE CONTAIN HUMAN KING?
FBA1 D0 02                      BNE     JUMPS           ;NO.
FBA3 09 80                      ORA     #$80            ;YES.  KING
FBA5                    JUMPS
FBA5 A6 B4                      LDX     T1
FBA7 95 C5                      STA     JMPLST,X
FBA9 E6 B4                      INC     T1

FBAB 68                         PLA                     ;RELOAD JUMP SQUARE
FBAC 18                         CLC                     ;COMPUTE TO SQUARE
FBAD 79 66 FF                   ADC     MOVTAB,Y
FBB0 85 B3                      STA     TO
FBB2 E4 A4                      CPX     JMPI            ;DONE?
FBB4 60                         RTS
```

```
                        ;
                        ;       JUMP4 -- SIMILAR TO JUMP1, EXCEPT DOES 4 JUMPS
                        ;
                        ;       INPUT:  A = 4 OFFSET INDEXES (B7-6 = FIRST)
                        ;               T0 = START SQUARE (FOR JUMP1)
                        ;               T1 = JMPLST INDEX (FOR JUMP1)
                        ;
                        ;       USES:   T2 = LOOP COUNT
                        ;               T3 = INDEXES
                        ;
                        ;       OUTPUT: T0 = END SQUARE, CS=> DONE
                        ;
FBB5                    JUMP4
FBB5 85 B6                      STA     T3
FBB7 A9 03                      LDA     #3
FBB9 85 B5                      STA     T2
FBBB                    JUMPL
FBBB A9 00                      LDA     #0
FBBD 06 B6                      ASL     T3              ;SHIFT 2 BITS OF INDEX FROM MSB'S OF T3 TO LSB'S OF A
FBBF 2A                         ROL     A
FBC0 06 B6                      ASL     T3
FBC2 2A                         ROL     A
FBC3 A8                         TAY
FBC4 20 80 FB                   JSR     JUMP1
FBC7 B0 04                      BCS     JUMPR           ;BRANCH IF DONE
FBC9 C6 B5                      DEC     T2
FBCB 10 EE                      BPL     JUMPL           ;CONTINUE
FBCD                    JUMPR
FBCD 60                         RTS
```

```
                        ;
                        ;       TERMNODE -- CHECK FOR TERMINAL NODE
                        ;
                        ;       RETURN ACTIVE SWITCHED
                        ;       CC => NOT TERMINAL --  XMOVE, AND MSTJMP SET UP FOR NEXT LEVEL
                        ;       CS => TERMINAL -- NUM = STATIC VALUE OF BOARD
FBCE                    TERMNODE
FBCE 20 14 FB                   JSR     SWIT            ;SWITCH ACTIVE PLAYER TO GET ACTIVE FOR NEXT LEVEL
FBD1 20 06 FA                   JSR     JMPCK2          ;CHECK FOR MOVE OR JUMP, SET UP MSTJMP, XMOVE,YMOVE
FBD4 D0 0F                      BNE     ALP200          ;CAN MOVE

FBD6 A6 BC                      LDX     DEPTH           ;CAN'T MOVE -- VERY BAD
FBD8 8A                         TXA
FBD9 18                         CLC
FBDA 69 02                      ADC     #2
FBDC 95 DF                      STA     ALPHAL+1,X
FBDE A9 80                      LDA     #$80
FBE0 95 EC                      STA     ALPHAH+1,X      ;-INFINITY = $8002+DEPTH = -(7FFF - DEPTH-1) (-7FFD TO -7FF3)  ($8003 - $800
FBE2 4C 3A FE                   JMP     SED20           ;NEGATE IF LOSING CHECKERS & RETURN CARRY SET
                        ;
                        ;       ALP200 -- ENTRY POINT -- CHECK DEPTH -- WE KNOW WE CAN JUMP.  CS=>TERMINAL NODE
                        ;
FBE5 A5 BC              ALP200  LDA     DEPTH
FBE7 C9 0B                      CMP     #MAXMAX
FBE9 90 22                      BCC     ALP220

FBEB 20 1D FC                   JSR     STATEVAL        ;"STACK" FULL -- MUST STOP
FBEE A5 A5                      LDA     MSTJMP
FBF0 F0 19                      BEQ     ALP210
                        ;                               MUST JUMP -- MODIFY VALUE FOR PENDING JUMP
                        ;                               DECREASE MAGNITUDE OF SCORE.
FBF2 A0 00                      LDY     #0              ;ADD NEGATIVE NUMBER
FBF4 A6 BC                      LDX     DEPTH
FBF6 B5 EC                      LDA     ALPHAH+1,X
FBF8 10 02                      BPL     ALP205
FBFA A0 02                      LDY     #2              ;<0  =>  ADD POSITIVE NUMBER
FBFC                    ALP205
FBFC B9 C8 FF                   LDA     ADDTAB,Y        ;POS.  => -ADDTRM.    NEG => ADDTRM
FBFF 18                         CLC
FC00 75 DF                      ADC     ALPHAL+1,X
FC02 95 DF                      STA     ALPHAL+1,X
FC04 B9 C9 FF                   LDA     ADDTAB+1,Y      ;POS=> $FF.   NEG => 0
FC07 75 EC                      ADC     ALPHAH+1,X
FC09 95 EC                      STA     ALPHAH+1,X

FC0B                    ALP210
FC0B 38                         SEC
FC0C 60                         RTS

FC0D                    ALP220
FC0D A6 A5                      LDX     MSTJMP          ;MUST JUMP?
FC0F F0 02                      BEQ     ALP235          ;NO.  BRANCH
FC11                    ALP230
FC11 18                         CLC                     ;YES.  CONTINUE TREE SEARCH
FC12                    ALP231
FC12 60                         RTS
FC13                    ALP235
FC13 A5 B0                      LDA     GAMNO           ;CHECK MAX DEPTH
FC15 29 0F                      AND     #$0F
FC17 C5 BC                      CMP     DEPTH
```

VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM

```
FC19 F0 02                  BEQ     ALP240          ;IF GAMNO = DEPTH THEN STOP UNLESS OTHER PLAYER CAN JUMP
FC1B B0 F4                  BCS     ALP230          ;IF GAMNO > DEPTH THEN CONTINUE TREE SEARCH
                            ;
                            ;                       ;IF GAMNO < DEPTH THEN STOP UNLESS OTHER PLAYER CAN JUMP
                            ;
FC1D                        ALP240
```

```
                        ;       STATEVAL -- COMPUTE STATIC VALUE OF BOARD
                        ;
                        ;       HUMAN ACTIVE => -CPIECE/HPIECE   THEN SHIFT LEFT 4 BITS  (2 BYTE RESULT)
                        ;       COMPUTER ACTIVE => CPIECE/HPIECE
                        ;       RESULT IS TWO BYTES IN NUM & ALPHA(DEPTH+1) (THIS MIGHT REPLACE OLD ALPHA VALUE).
                        ;       NUM = LSB = FRACTIONAL PART.
                        ;       NUM+1 = MSB = INTEGER PART.
                        ;
                        ;       RETURN CARRY SET
                        ;
FC1D                    STATEVAL
FC1D A6 A9                      LDX     HPIECE
FC1F A4 AA                      LDY     CPIECE
FC21 86 B8                      STX     DEN             ;HUMAN ACTIVE => HPIECE/CPIECE
FC23 84 B6                      STY     NUM+1           ;MSB
FC25 A9 00                      LDA     #0              ;LSB
FC27 85 B5                      STA     NUM
FC29 A2 11                      LDX     #17
FC2B 18                         CLC
FC2C                    SELP
FC2C 26 B5                      ROL     NUM             ;LSB
FC2E 26 B6                      ROL     NUM+1           ;MSB
FC30 CA                         DEX
FC31 F0 09                      BEQ     SEDONE          ;BRANCH IF DONE
FC33 2A                         ROL     A
FC34 C5 B8                      CMP     DEN
FC36 90 F4                      BCC     SELP            ;0
FC38 E5 B8                      SBC     DEN
FC3A B0 F0                      BCS     SELP            ;JUMP -- ROTATE IN 1
FC3C                    SEDONE
FC3C A9 00                      LDA     #0
FC3E 85 B4                      STA     T1              ;INIT CREDIT TO 0

FC40 06 B5                      ASL     NUM             ;SHIFT LEFT 4 BITS
FC42 26 B6                      ROL     NUM+1           ;CHECK FOR OVERFLOW
FC44 06 B5                      ASL     NUM
FC46 26 B6                      ROL     NUM+1
FC48 06 B5                      ASL     NUM
FC4A 26 B6                      ROL     NUM+1
FC4C 06 B5                      ASL     NUM
FC4E 26 B6                      ROL     NUM+1
FC50 B0 02                      BCS     SED05           ;OVERFLOW
FC52 10 0B                      BPL     SED07           ;NO OVERFLOW
FC54                    SED05
FC54 A9 F2                      LDA     #$FF-MAXMAX-2   ;$F2 IF MAXMAX = 11
FC56 85 B5                      STA     NUM
FC58 A9 7F                      LDA     #$7F            ;SET TO MAX MAGNITUDE (-$800E TO ALLOW ROOM FOR DEC)
FC5A 85 B6                      STA     NUM+1
FC5C 4C 1A FE                   JMP     ADDSUM          ;IGNORE REST OF CREDITS
FC5F                    SED07
```

```
                        ;
                        ;       END GAME CLEAN UP -- DO ONLY IF COMPUTER IS WINNING OR EVEN
                        ;
FC5F A9 FF                      LDA     #$FF
FC61 85 B9                      STA     T7              ;INDICATE NOT END GAME UNTIL PROVEN OTHERWISE

FC63 A5 A9                      LDA     HPIECE
FC65 38                         SEC                     ;COMPUTE # OF HUMAN PIECES * 2 = HPIECE-HKING
FC66 E5 AE                      SBC     HKING
FC68 C9 08                      CMP     #ENDPC          ;DOES HUMAN HAVE MORE THAN END GAME AMOUNT?
FC6A 90 03                      BCC     DIS03
FC6C                    DIS02
FC6C 4C 10 FD                   JMP     DIS70           ;YES.  NOT END GAME
FC6F                    DIS03
FC6F 65 AF                      ADC     CKING           ;NO.  END GAME.  IS COMPUTER AHEAD IN # OF PIECES?
FC71 38                         SEC
FC72 E5 AA                      SBC     CPIECE
FC74 B0 F6                      BCS     DIS02           ;SKIP IF HUMAN >= COMPUTER

FC76                    DIS04
FC76 A2 22                      LDX     #34
FC78                    DISLP1
FC78 86 B9                      STX     T7              ;SAVE X
FC7A B5 80                      LDA     BOARD,X         ;EMPTY SQUARE?
FC7C F0 64                      BEQ     DIS65           ;YES.  TRY NEXT SQUARE

FC7E 29 20                      AND     #KING           ;KING?
FC80 D0 0D                      BNE     DIS05           ;YES.  SKIP

FC82 20 7D FA                   JSR     XIE1            ;NO.  ORDINARY MAN.  COMPUTE DISTANCE OF MAN FROM BOTTOM OF BOARD
FC85 8A                         TXA
FC86 4A                         LSR     A
FC87 4A                         LSR     A
FC88 18                         CLC
FC89 65 B4                      ADC     T1              ;ADD T1
FC8B 85 B4                      STA     T1
FC8D A6 B9                      LDX     T7              ;RELOAD X
FC8F                    DIS05
FC8F B5 80                      LDA     BOARD,X
FC91 30 4F                      BMI     DIS65           ;SKIP IF COMPUTER
FC93 20 7D FA                   JSR     XIE1            ;CONVERT INTERNAL SQUARE # TO EXTERNAL
FC96 8A                         TXA
FC97 4A                         LSR     A
FC98 4A                         LSR     A
FC99 85 B7                      STA     T6              ;ROW
FC9B 49 01                      EOR     #1              ;COMPLEMENT LSBIT
FC9D 6A                         ROR     A               ;AND SHIFT TO CARRY.
FC9E 8A                         TXA
FC9F 2A                         ROL     A               ;SHIFT CARRY INTO B0
FCA0 29 07                      AND     #7              ;AND MASK TO GET COL.
FCA2 85 B8                      STA     DEN             ;COL

FCA4 A2 22                      LDX     #34             ;FIND COMPUTER'S PIECES
FCA6                    DISLP2
FCA6 B5 80                      LDA     BOARD,X
FCA8 10 35                      BPL     DIS60           ;EMPTY OR HUMAN
                        ;                               CHECK FOR KING?
FCAA 8A                         TXA                     ;SAVE INTERNAL SQUARE #
FCAB 48                         PHA
FCAC 20 7D FA                   JSR     XIE1            ;CONVERT INTERNAL SQUARE # TO EXTERNAL
```

Atari Business Game, Inc.

```
FCAF 8A                     TXA
FCB0 4A                     LSR     A               ;COMPUTE ROW
FCB1 4A                     LSR     A
FCB2 38                     SEC
FCB3 E5 B7                  SBC     T6              ;COMPUTE DIFFERENCE IN ROWS
FCB5 10 05                  BPL     DIS30
FCB7 49 FF                  EOR     #$FF            ;MAKE POSITIVE
FCB9 18                     CLC
FCBA 69 01                  ADC     #1
FCBC                DIS30
FCBC 85 B3                  STA     T0
FCBE 8A                     TXA                     ;COMPUTE DIFFERENCE IN COLS
FCBF 4A                     LSR     A
FCC0 4A                     LSR     A
FCC1 49 01                  EOR     #1
FCC3 6A                     ROR     A               ;SHIFT TO CARRY
FCC4 8A                     TXA
FCC5 2A                     ROL     A
FCC6 29 07                  AND     #7              ;COL
FCC8 38                     SEC
FCC9 E5 B8                  SBC     DEN
FCCB 10 05                  BPL     DIS40
FCCD 49 FF                  EOR     #$FF            ;MAKE POSITIVE
FCCF 18                     CLC
FCD0 69 01                  ADC     #1
FCD2                DIS40
FCD2 C5 B3                  CMP     T0              ;FIND LARGER OF DIF IN ROWS AND COLS
FCD4 B0 02                  BCS     DIS50
FCD6 A5 B3                  LDA     T0              ;USE FIRST VALUE
FCD8                DIS50
FCD8 18                     CLC                     ;THIS IS THE DISTANCE BET. THE COMPUTER PIECE & THE HUMAN PIECE
FCD9 65 B4                  ADC     T1              ;ADD NEG. # TO CREDIT
FCDB 85 B4                  STA     T1
FCDD 68                     PLA                     ;RESTORE INTERNAL SQUARE #
FCDE AA                     TAX
FCDF                DIS60
FCDF CA                     DEX
FCE0 10 C4                  BPL     DISLP2

FCE2                DIS65
FCE2 A6 B9                  LDX     T7              ;STOP WHEN T7=0
FCE4 CA                     DEX
FCE5 10 91                  BPL     DISLP1

FCE7 A5 B5                  LDA     NUM             ;SUBTRACT CREDIT FROM NUM
FCE9 38                     SEC
FCEA E5 B4                  SBC     T1
FCEC 85 B5                  STA     NUM
FCEE B0 02                  BCS     DIS67           ;BORROW?
FCF0 C6 B6                  DEC     NUM+1           ;YES. DECREMENT MSB
FCF2                DIS67
FCF2 A9 00                  LDA     #0              ;CLEAR OUT T1
FCF4 85 B4                  STA     T1
```

```
                        ;
                        ;       COUNT COMPUTER PIECES ON DIAGONAL
                        ;
FCF6 A0 0D                      LDY     #14-1
FCF8                    DIAGLP
FCF8 BE 7E FF                   LDX     DIAG,Y
FCFB B5 80                      LDA     BOARD,X
FCFD F0 0E                      BEQ     DIAG20          ;EMPTY
FCFF 2A                         ROL     A
FD00 29 40                      AND     #KING*2         ;KING?
FD02 F0 09                      BEQ     DIAG20          ;NO.
FD04 90 05                      BCC     DIAG10          ;YES.  BRANCH IF HUMAN
FD06 E6 B4                      INC     T1              ;COMPUTER
FD08 4C 0D FD                   JMP     DIAG20
FD0B                    DIAG10
FD0B C6 B4                      DEC     T1              ;HUMAN
FD0D                    DIAG20
FD0D 88                         DEY
FD0E 10 E8                      BPL     DIAGLP
FD10                    DIS70
```

```
                        ;
                        ;       COMPUTE BACK ROW CREDIT (KING ROW PROTECTION)
                        ;
FD10 A5 B0                      LDA     GAMNO
FD12 29 0E                      AND     #$0E            ;GAME # = 1?
FD14 D0 03                      BNE     ADS10           ;NO.  SKIP
FD16                BACK25
FD16 4C 1A FE                   JMP     ADDSUM          ;SKIP THE REST IF GAME # = 1 OR $11
FD19                ADS10

FD19 A5 AE                      LDA     HKING           ;DOES HKING * 3 = HPIECE?
FD1B 0A                         ASL     A               ;SHIFT LEFT AND CLEAR CARRY
FD1C 65 AE                      ADC     HKING
FD1E C5 A9                      CMP     HPIECE
FD20 F0 23                      BEQ     BACK10          ;YES.   => NO ORDINARY MEN TO CONVERT TO KINGS

FD22 A9 00                      LDA     #0
FD24 A4 9F                      LDY     BOARD+31
FD26 10 02                      BPL     BACK30
FD28 09 08                      ORA     #8              ;OR IN 1 BIT FOR EACH BACK ROW SQUARE WHICH IS OCCUPIED BY A COMP PIECE
FD2A                BACK30
FD2A A4 A0                      LDY     BOARD+32
FD2C 10 02                      BPL     BACK40
FD2E 09 04                      ORA     #4
FD30                BACK40
FD30 A4 A1                      LDY     BOARD+33
FD32 10 02                      BPL     BACK50
FD34 09 02                      ORA     #2
FD36                BACK50
FD36 A4 A2                      LDY     BOARD+34
FD38 10 02                      BPL     BACK60
FD3A 09 01                      ORA     #1
FD3C                BACK60
FD3C AA                         TAX
FD3D BD 52 FF                   LDA     BAKTAB,X        ;ADD TABLE VALUE TO CURRENT CREDIT
FD40 18                         CLC
FD41 65 B4                      ADC     T1
FD43 85 B4                      STA     T1

FD45                BACK10
FD45 A5 AF                      LDA     CKING           ;DOES CKING*3 = CPIECE?
FD47 0A                         ASL     A
FD48 65 AF                      ADC     CKING
FD4A C5 AA                      CMP     CPIECE
FD4C F0 2B                      BEQ     BACK20          ;YES.   NO CHECKERS TO CONVERT TO KINGS

FD4E A9 00                      LDA     #0              ;OR IN 1 BIT FOR EACH BACK ROW SQUARE WHICH IS OCCUPIED BY A HUMAN PIECE
FD50 A4 83                      LDY     BOARD+3
FD52 F0 04                      BEQ     BACK70
FD54 30 02                      BMI     BACK70
FD56 09 08                      ORA     #8
FD58                BACK70
FD58 A4 82                      LDY     BOARD+2
FD5A F0 04                      BEQ     BACK80
FD5C 30 02                      BMI     BACK80
FD5E 09 04                      ORA     #4
FD60                BACK80
FD60 A4 81                      LDY     BOARD+1
FD62 F0 04                      BEQ     BACK90
FD64 30 02                      BMI     BACK90
```

VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM

```
FD66 09 02                      ORA     #2
FD68                    BACK90
FD68 A4 80                      LDY     BOARD+0
FD6A F0 04                      BEQ     BACK95
FD6C 30 02                      BMI     BACK95
FD6E 09 01                      ORA     #1
FD70                    BACK95
FD70 AA                         TAX
FD71 A5 B4                      LDA     T1
FD73 38                         SEC
FD74 FD 52 FF                   SBC     BAKTAB,X        ;SUBTRACT TABLE VALUE FROM CURRENT CREDIT
FD77 85 B4                      STA     T1

FD79                    BACK20
```

```
                        ;
                        ;       COMPUTE DOUBLE CORNER CREDIT
                        ;
FD79 A5 B0                      LDA     GAMNO
FD7B 29 0F                      AND     #$0F
FD7D C9 03                      CMP     #3              ;GAME # = 1 OR 2?
FD7F 90 95                      BCC     BACK25          ;JMP ADDSUM IF GAME # TOO SMALL

FD81 A9 20                      LDA     #HUMKNG
FD83 A2 FC                      LDX     #-CORN&$FF
FD85 A4 A9                      LDY     HPIECE
FD87 C4 AA                      CPY     CPIECE
FD89 F0 06                      BEQ     CORN20          ;DO IF EQUAL
FD8B 90 04                      BCC     CORN20          ;HUMAN <= COMPUTER ==> HUMAN WANTS CORN
FD8D A9 A0                      LDA     #CMPKNG         ;HUMAN > COMPUTER =>COMPUTER WANTS CORN
FD8F A2 04                      LDX     #CORN           ;INC ALPHA
FD91                CORN20
FD91 85 B8                      STA     DEN             ;HUMAN OR COMPUTER
FD93 86 B3                      STX     TO              ;-CORN OR +CORN
FD95 A0 00                      LDY     #0
FD97 A5 80                      LDA     BOARD+0         ;IS SQUARE OCCUPIED?
FD99 F0 05                      BEQ     CORN30          ;NO.  EMPTY
FD9B 45 B8                      EOR     DEN             ;CORRECT PLAYER AND KING?
FD9D D0 15                      BNE     CORN50          ;NO.  OTHER PLAYER -- TOO BAD
FD9F C8                         INY                     ;YES.  INC CORN COUNT
FDA0                CORN30
FDA0 A5 84                      LDA     BOARD+4         ;CHECK OTHER SQUARE IN THIS CORNER
FDA2 F0 05                      BEQ     CORN40          ;EMPTY
FDA4 45 B8                      EOR     DEN
FDA6 D0 0C                      BNE     CORN50          ;WRONG PLAYER -- TOO BAD
FDA8 C8                         INY
FDA9                CORN40
FDA9 C0 01                      CPY     #1
FDAB D0 07                      BNE     CORN50
FDAD A5 B4                      LDA     T1
FDAF 18                         CLC
FDB0 65 B3                      ADC     TO              ;WE GET CREDIT
FDB2 85 B4                      STA     T1
FDB4                CORN50
FDB4 A0 00                      LDY     #0              ;CHECK OTHER CORNER
FDB6 A5 9E                      LDA     BOARD+30        ;IS SQUARE OCCUPIED?
FDB8 F0 05                      BEQ     CORN60          ;NO.  EMPTY
FDBA 45 B8                      EOR     DEN             ;CORRECT PLAYER AND KING?
FDBC D0 15                      BNE     CRN100          ;NO.  TOO BAD
FDBE C8                         INY
FDBF                CORN60
FDBF A5 A2                      LDA     BOARD+34
FDC1 F0 05                      BEQ     CORN70          ;NO.  EMPTY
FDC3 45 B8                      EOR     DEN             ;CORRECT PLAYER AND KING?
FDC5 D0 0C                      BNE     CRN100          ;NO.
FDC7 C8                         INY
FDC8                CORN70
FDC8 C0 01                      CPY     #1
FDCA D0 07                      BNE     CRN100
FDCC A5 B4                      LDA     T1
FDCE 18                         CLC
FDCF 65 B3                      ADC     TO
FDD1 85 B4                      STA     T1
FDD3                CRN100
```

NOP NOP?

```
                        ;
                        ;       COMPUTE CENTER CREDIT
                        ;
FDD3                    DOCENT
FDD3 A5 B9                      LDA     T7
FDD5 F0 0C                      BEQ     CENT90          ;SKIP IF END GAME
FDD7 A5 93                      LDA     BOARD+19        ;COUNT EACH COMPUTER PIECE IN SQUARES 19&20
FDD9 10 02                      BPL     CENT50
FDDB E6 B4                      INC     T1
FDDD                    CENT50
FDDD A5 94                      LDA     BOARD+20
FDDF 10 02                      BPL     CENT90
FDE1 E6 B4                      INC     T1
FDE3                    CENT90
```

```
                        ;
                        ;       COMPUTE MOVE CREDIT (ADD CREDIT IF COMPUTER HAS MOVE)
                        ;
FDE3 A5 B0                      LDA     GAMNO
FDE5 29 0C                      AND     #$0C            ;GAME # < 4?
FDE7 F0 31                      BEQ     ADDSUM          ;YES. SKIP
FDE9 A5 A9                      LDA     HPIECE
FDEB 18                         CLC
FDEC 65 AA                      ADC     CPIECE
FDEE C9 19                      CMP     #24+1           ;TOTAL PIECE COUNT > 24?
FDF0 B0 28                      BCS     QMV100          ;YES. SKIP MOVE

FDF2 A5 A9                      LDA     HPIECE          ;SUBTRACT OFF EXTRA POINT FOR EACH KING TO GET
FDF4 38                         SEC                     ;2 TIMES PIECE COUNT FOR EACH PLAYER.
FDF5 E5 AE                      SBC     HKING
FDF7 38                         SEC
FDF8 E5 AA                      SBC     CPIECE          ;HPIECE-HKING-CPIECE+CKING
FDFA 18                         CLC                     ;=HPIECE-HKING-(CPIECE-CKING)
FDFB 65 AF                      ADC     CKING
FDFD D0 1B                      BNE     QMV100          ;PIECE COUNTS ARE DIFFERENT

                        ;
                        ;       COUNT PIECES IN COMPUTER'S SYSTEM
                        ;
FDFF A0 0F                      LDY     #16-1
FE01 A9 00                      LDA     #0              ;INIT COUNT
FE03 85 B3                      STA     T0
FE05                    QMVLP
FE05 BE 75 FF                   LDX     SYSTEM,Y
FE08 B5 80                      LDA     BOARD,X
FE0A F0 02                      BEQ     QMV10
FE0C E6 B3                      INC     T0
FE0E                    QMV10
FE0E 88                         DEY
FE0F 10 F4                      BPL     QMVLP

FE11 66 B3                      ROR     T0              ;ODD OR EVEN?
FE13 6A                         ROR     A
FE14 45 BB                      EOR     ACTIVE          ;COMPUTER OR HUMAN ACTIVE?
FE16 30 02                      BMI     QMV100          ;HUMAN HAS MOVE -- DON'T ADD CREDIT
FE18 E6 B4                      INC     T1              ;HUMAN IS ACTIVE AND EVEN COUNT
FE1A                    QMV100
```

```
                        ;
                        ;       ADD SUM OF BACK, CORN, ETC.  IN T1 TO PIECE COUNT RATIO IN NUM
                        ;
FE1A                    ADDSUM
FE1A A6 BC                      LDX     DEPTH
FE1C A5 B4                      LDA     T1
FE1E 18                         CLC
FE1F 65 B5                      ADC     NUM
FE21 95 DF                      STA     ALPHAL+1,X
FE23 A9 00                      LDA     #0              ;COMPUTE MSB
FE25 A4 B4                      LDY     T1              ;NEGATIVE?
FE27 10 02                      BPL     ADD10           ;NO.  POSITIVE.  USE 0
FE29 A9 FF                      LDA     #$FF            ;YES.   USE $FF
FE2B                    ADD10
FE2B 65 B6                      ADC     NUM+1           ;ADD WITH CARRY FROM LSB
FE2D 95 EC                      STA     ALPHAH+1,X      ;SHOULD NEVER GET OVERFLOW

FE2F A5 BB                      LDA     ACTIVE          ;HUMAN ACTIVE?
FE31 30 07                      BMI     SED20           ;NO.
FE33 A5 B0                      LDA     GAMNO
FE35 C9 10                      CMP     #PLAY2          ;LOSING CHECKERS?
FE37 90 09                      BCC     SED25           ;NO.  WINNING CHECKERS -- NEGATE ALPHA
FE39 60                         RTS                     ;YES.  DOUBLE NEGATE => NO NEGATE.   RETURN CARRY SET
                        ;
                        ;       IF LOSING CHECKERS (GAMES > 2-PLAYER GAME) THEN NEGATE ALPHA -- ENTRY POINT
                        ;
                        ;       RETURN CARRY SET
                        ;
FE3A                    SED20
FE3A A6 BC                      LDX     DEPTH
FE3C A5 B0                      LDA     GAMNO
FE3E C9 10                      CMP     #PLAY2
FE40 90 0D                      BCC     SED30
FE42                    SED25
FE42 38                         SEC                     ;LOSING CHECKERS
FE43 A9 00                      LDA     #0
FE45 F5 DF                      SBC     ALPHAL+1,X
FE47 95 DF                      STA     ALPHAL+1,X
FE49 A9 00                      LDA     #0
FE4B F5 EC                      SBC     ALPHAH+1,X
FE4D 95 EC                      STA     ALPHAH+1,X
FE4F                    SED30
FE4F 38                         SEC                     ;RETURN CARRY SET (FOR TERMNODE)
FE50 60                         RTS
```

```
                        ;
                        ;       CHGCNT -- CHANGE PIECE COUNT TO REFLECT OPPONENT'S JUMPED PIECE
                        ;
FE51                    CHGCNT
FE51 A0 FE                      LDY     #-2&$FF         ;DECREASE OPPONENT'S PIECE COUNT FOR JUMPED PIECE
FE53 AA                         TAX                     ;KING?
FE54 10 01                      BPL     AJ10            ;NO.
FE56 88                         DEY                     ;YES. USE -3
FE57                    AJ10
FE57 A5 BB                      LDA     ACTIVE
FE59 49 80                      EOR     #$80            ;OPPONENT
FE5B 0A                         ASL     A
FE5C 2A                         ROL     A               ;SHIFT CLEARS CARRY
FE5D AA                         TAX
FE5E 98                         TYA
FE5F 75 A9                      ADC     HPIECE,X
FE61 95 A9                      STA     HPIECE,X

FE63 C0 FD                      CPY     #-3&$FF
FE65 D0 02                      BNE     AJ15
FE67 D6 AE                      DEC     HKING,X         ;DECREMENT KING COUNT BECAUSE KING WAS CAPTURED
FE69                    AJ15
FE69 60                         RTS
                        ;
                        ;       JMPSV2 -- SAVE MOVE IN JMPLST (COMPACTED FORMAT FOR ALPHA-BETA)
                        ;
FE6A                    JMPSV2
FE6A A0 00                      LDY     #0              ;YES. SAVE FIRST MOVE IN JMPLST
FE6C A6 BC                      LDX     DEPTH           ;1-9
FE6E CA                         DEX
FE6F D0 0A                      BNE     WASJ18

FE71 A5 BE                      LDA     INDEX           ;DEPTH=1
FE73 4A                         LSR     A
FE74 6A                         ROR     A
FE75 6A                         ROR     A
FE76 05 AC                      ORA     FROMB
FE78 85 C5                      STA     JMPLST
FE7A 60                         RTS
FE7B                    WASJ18
FE7B CA                         DEX                     ;DEPTH-2  (0-7)
FE7C 8A                         TXA
FE7D 48                         PHA
FE7E 4A                         LSR     A
FE7F 4A                         LSR     A               ;0 OR 1
FE80 AA                         TAX
FE81 E8                         INX                     ;1 OR 2
FE82 68                         PLA
FE83 29 03                      AND     #3              ;0-2
FE85 A8                         TAY

FE86 A5 BE                      LDA     INDEX
FE88 84 B3                      STY     TO
FE8A                    WASJLP
FE8A C0 03                      CPY     #3              ;DONE?
FE8C B0 05                      BCS     WASJ90          ;YES.
FE8E 0A                         ASL     A               ;NO.   SHIFT LEFT 2 MORE BITS
FE8F 0A                         ASL     A
FE90 C8                         INY
FE91 90 F7                      BCC     WASJLP          ;JMP
```

```
FE91 90 F7                      BCC     WASJLP          ;JMP
```



```
                FE93            WASJ90
FE93 A4 B3                      LDY     TO
FE95 55 C5                      EOR     JMPLST,X
FE97 39 4A FF                   AND     MSKLST,Y        ;COMBINE WITH JMPLST (C0,30,0C,03)
FE9A 55 C5                      EOR     JMPLST,X
FE9C 95 C5                      STA     JMPLST,X
FE9E 60                         RTS
                ;
                ;       JMPSAV -- SAVE VARIABLES FOR JUMP & INC DEPTH
                ;
                FE9F            JMPSAV
FE9F A5 BD                      LDA     JMPSQR
FEA1 29 C0                      AND     #$C0            ;JUMPED PIECE
FEA3 05 A7                      ORA     MSTFLG
FEA5 05 A8                      ORA     MLTFLG
FEA7 05 BE                      ORA     INDEX
FEA9 45 A3                      EOR     PIECE
FEAB 29 CF                      AND     #$CF            ;COMBINE A WITH BITS 5-4 OF PIECE (KING OR CHECKER)
FEAD 45 A3                      EOR     PIECE
                ;
                ;       SAVE1 -- ENTRY POINT
                ;
                FEAF            SAVE1
FEAF E6 BC                      INC     DEPTH
FEB1 A6 BC                      LDX     DEPTH
FEB3 95 D1                      STA     OTHER-2,X
FEB5 A5 AC                      LDA     FROMB
FEB7 05 BF                      ORA     FRSMOV
FEB9 95 C6                      STA     FROMT-2,X
FEBB 60                         RTS
```

```
                        ;
                        ;       KNGCK2 -- RETURN CS IF NEW KING ON SQUARE X(0-34), A=CONTENTS OF SQUARE X
                        ;                 INC CPIECE OR HPIECE IF NEW KING
                        ;
FEBC                    KNGCK2
FEBC C9 90                      CMP     #CMPCHK
FEBE D0 0A                      BNE     TO60
FEC0 E0 04                      CPX     #4              ;COMPUTER CHECKER TO KING IF ON 0-3
FEC2 B0 18                      BCS     TO80            ;NOT NEW KING
FEC4 E6 AA                      INC     CPIECE          ;INCREMENT COMPUTER PIECE COUNT
FEC6 E6 AF                      INC     CKING           ;INC KING COUNT
FEC8 90 0C                      BCC     TO70            ;JMP    NEW KING
FECA                    TO60
FECA C9 10                      CMP     #HUMCHK
FECC D0 0E                      BNE     TO80
FECE E0 1F                      CPX     #31             ;HUMAN CHECKER TO KING IF ON 31-34
FED0 90 0A                      BCC     TO80
FED2 E6 A9                      INC     HPIECE          ;INCREMENT HUMAN PIECE COUNT FOR NEW KING
FED4 E6 AE                      INC     HKING           ;INC KING COUNT
FED6                    TO70
FED6 29 80                      AND     #$80
FED8 09 20                      ORA     #KING
FEDA 38                         SEC
FEDB 60                         RTS
FEDC                    TO80
FEDC 18                         CLC
FEDD 60                         RTS
                        ;
                        ;       TOMOV -- CALL TOSET, MOVCLR
                        ;
FEDE                    TOMOV
FEDE 20 E2 FA                   JSR     TOSET
                        ;
                        ;       MOVCLR -- SET MOVE TO 8 (OFF BOARD SQUARE), SET MOVVAL TO 0
                        ;                 RETURN Y=0
                        ;
FEE1                    MOVCLR
FEE1 A0 08                      LDY     #8
FEE3 84 E9                      STY     MOVE
FEE5 A0 00                      LDY     #0
FEE7 84 E4                      STY     MOVVAL
FEE9 60                         RTS

                        ;                       RANDOM MOVE TABLE -- INDEXED BY PRNCNT
                        ;                   1/2, 1/3, 1/4, 1/5, 1/6, 1/7
FEEA 80 55 40           RANTAB  .BYTE   $80, $55, $40, $33, $2B, $25
FEED 33 2B 25

FEF0 00                         .BYTE   0               ;EXTRA BYTE
                                .IF     PAL-1
FEF1 00 00 00                   .BYTE   0, 0, 0, 0, 0, 0, 0, 0 ;MORE EXTRA BYTES IF NOT PAL
FEF4 00 00 00
FEF7 00 00
FEF9 00 00 00                   .BYTE   0, 0, 0, 0, 0, 0, 0
FEFC 00 00 00
FEFF 00
                                .ENDIF
```

```
                                .IF     PRNT
                          AD=*
                                .ENDIF
```

```
                        ;                               DATA
                                *=ROMSTR+$F00
                        ;                                       BOARD CHARACTERS (IN UPSIDE DOWN ORDER)
        FF00            BRDCHR
        FF00 00 00 00           .BYTE   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0                                 ;EMPTY 0*LINCHR
        FF03 00 00 00
        FF06 00 00 00
        FF09 00 00 00
        FF0C 00 00 00
        FF0F 00
        FF10 3C 3C 7E           .BYTE   $3C,$3C,$7E,$7E,$FF,$FF,$FF,$FF,$FF,$FF,$FF,$FF,$7E,$7E,$3C,$3C ;CHECKER 1
        FF13 7E FF FF
        FF16 FF FF FF
        FF19 FF FF FF
        FF1C 7E 7E 3C
        FF1F 3C
        FF20 3C 3C 7E           .BYTE   $3C,$3C,$7E,$7E,$C3,$C3,$C3,$C3,$A5,$A5,$A5,$E7,$7E,$7E,$3C,$3C ;KING 2
        FF23 7E C3 C3
        FF26 C3 C3 A5
        FF29 A5 A5 E7
        FF2C 7E 7E 3C
        FF2F 3C
        FF30 00 00 00           .BYTE   0,0,0,$C6,$C6,$6C,$6C,$38,$38,$6C,$6C,$C6,$C6,0         ;CURSOR 3
        FF33 C6 C6 6C
        FF36 6C 38 38
        FF39 6C 6C C6
        FF3C C6 00
        FF3E 00 00      HTAB    .BYTE   0,0                     ;CHANGES FOR HPIECE (BASED ON NXTAB). 0,0 IS PART OF CURSOR
V       FF40 FD 01 02   CTAB    .BYTE   -3,1,2                  ;PART OF HTAB
        FF43 00 00      HKTAB   .BYTE   0,0                     ;CHANGES FOR HKING (KING COUNT). 0,0 IS PART OF CTAB
V       FF45 FF 01 00   CKTAB   .BYTE   -1,1,0,0,0              ;CHANGES FOR CKING. -1,1,0 IS PART OF HKTAB
        FF48 00 00
        FF4A C0 30 0C   MSKLST  .BYTE   $C0,$30,$0C,$03         ;MASKS FOR DIFFERENT 2 BIT COMBOS -- USED IN JMPSV2
        FF4D 03
                        ;                                       NEXT CURSOR VALUES FOR SETUP
        FF4E A0 90 20   NXTAB   .BYTE   CMPKNG,CMPCHK,HUMKNG,HUMCHK     ;EMPTY
        FF51 10
                        ;       BAKTAB -- LIST OF BACK ROW CREDITS
                        ;       0=>     NO PIECE IN BACK ROW
                        ;       1=>     1 PIECE IN BACK ROW (OR 31 AND 32)
                        ;       2=>     2 PIECES IN BACK ROW (NOT 34)
                        ;       3=>     ANY 2 AS LONG AS 34
                        ;       4=>     ALL BUT 34
                        ;       5=>     ALL BUT 32
                        ;       6=>     32 AND 34 (AND MAYBE OTHERS)
                        ;
                        ;       FIRST 0 IS PART OF NXTAB
                        ;       SELECT BY DETERMINING 31,32,33,34 OR 3,2,1,0
```

```
FF52 00 02 02       BAKTAB  .BYTE   0*2,1*2,1*2,3*2,1*2,6*2,2*2,6*2,1*2,3*2,4*2,5*2,1*2,6*2,4*2,6*2
FF55 06 02 0C
FF58 04 0C 02
FF5B 06 08 0A
FF5E 02 0C 08
FF61 0C
FF62 05 09 06       JOYTAB  .BYTE   5,9,6,$A                ;JOYSTICK READINGS FOR DIAGONAL MOVEMENT.
FF65 0A
                    ;               MOVTAB & MVTAB2 GO TOGETHER -- DO NOT SEPARATE!!!
                    ;       HUMAN CHECKER CAN'T DO 0,1; COMPUTER CHECKER CAN'T DO 2,3.
                    ;
VV  FF66 FB FC 04   MOVTAB  .BYTE   RB,LB,RF,LF             ;OFFSETS FOR DIAGONAL MOVEMENT (RELATED TO JOYTAB)
FF69 05
VV  FF6A F6 F8 08   MVTAB2  .BYTE   2*RB,2*LB,2*RF,2*LF     ;OFFSETS FOR JUMPS
FF6D 0A
                    ;       LIST OF FIRST MOVES
FF6E 59 19 58       FRSTAB  .BYTE   $40+25,$00+25,$40+24,$00+24,$40+23,$40+22,$40+23        ;USE 23 FROM SYSTEM
FF71 18 57 56
FF74 57
                    ;       COMPUTER'S MOVE SYSTEM (INTERNAL NUMBERING)
FF75 17 06 07       SYSTEM  .BYTE   23,6,7,13,16,22,31,32,33        ;23 IS PART OF FRSTAB
FF78 0D 10 16
FF7B 1F 20 21
                    ;       SQUARES ON DIAGONAL BETWEEN DOUBLE CORNERS
FF7E 04 05 0E       DIAG    .BYTE   4,5,14,15,24,25,34                      ;PART OF SYSTEM ALSO
FF81 0F 18 19
FF84 22
FF85 00 09 0A               .BYTE   0,9,10,19,20,29,30                      ;REST OF DIAG
FF88 13 14 1D
FF8B 1E
                    ;       "SCORE" TABLE -- CHARACTERS 0-9, BLANK, JP
                    ;       MUST NOT CROSS PAGE BOUNDARY -- THIS INCLUDES BOTH LISTING PAGE & RAM PAGE.
FF8C                SCRTBL
FF8C 0E 0A 0A               .BYTE   $0E,$0A,$0A,$0A,$0E     ;0    HIGH ORDER 0 IS DISPLAYED AS BLANK
FF8F 0A 0E
FF91 EE 44 44               .BYTE   $EE,$44,$44,$CC,$44     ;1
FF94 CC 44
FF96 EE 88 EE               .BYTE   $EE,$88,$EE,$22,$EE     ;2
FF99 22 EE
FF9B EE 22 66               .BYTE   $EE,$22,$66,$22,$EE     ;3
FF9E 22 EE
FFA0 22 22 EE               .BYTE   $22,$22,$EE,$AA,$AA     ;4
FFA3 AA AA
FFA5 EE 22 EE               .BYTE   $EE,$22,$EE,$88,$EE     ;5
FFA8 88 EE
FFAA EE AA EE               .BYTE   $EE,$AA,$EE,$88,$EE     ;6
FFAD 88 EE
FFAF 22 22 22               .BYTE   $22,$22,$22,$22,$EE     ;7
FFB2 22 EE
FFB4 EE AA EE               .BYTE   $EE,$AA,$EE,$AA,$EE     ;8
FFB7 AA EE
FFB9 EE 22 EE               .BYTE   $EE,$22,$EE,$AA,$EE     ;9
FFBC AA EE
FFBE 00 00 00               .BYTE   0,0,0,0,0               ;AA     BLANK
FFC1 00 00
FFC3 E4 A4 27               .BYTE   $E4,$A4,$27,$25,$27     ;BB     JP
FFC6 25 27
```

VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM

```
V    FFC8 E0 FF 20      ADDTAB   .BYTE   -ADDTRM,$FF,ADDTRM      ;TERMS TO ADD IN TERMNODE IF FULL & PENDING JMP
                        ;                 USE 0 FROM COLTB FOR END OF ADDTAB
     FFCB 00 34 36      COLTB    .BYTE   EDARK,ELIGHT,ELIAT,HDARK,HLIGHT,HLIAT          ;COLORS FOR DIFFERENT SQUARES & EASY OR HARD
     FFCE B0 26 26
```

```
                        ;
                        ;       TOSUB -- SUBROUTINE TO DO JUNK FOR TO CODE
                        ;
FFD1                    TOSUB
FFD1 A6 E9                      LDX     MOVE            ;MOVE TO NEW SQUARE FOR MOVCHK
FFD3 A5 A3                      LDA     PIECE
FFD5 20 DF FF                   JSR     KNGCHK          ;NEW KING?
FFD8 85 A3                      STA     PIECE
FFDA 95 80                      STA     BOARD,X
FFDC 85 E4                      STA     MOVVAL
FFDE 60                         RTS
                        ;
                        ;       KNGCHK -- SAME AS KNGCK2 EXCEPT USES Y TO SET MSTJMP TO 0 IF NEW KING
                        ;
FFDF                    KNGCHK
FFDF 20 BC FE                   JSR     KNGCK2
FFE2 90 04                      BCC     TO81
FFE4 A0 00                      LDY     #0
FFE6 84 A5                      STY     MSTJMP
FFE8                    TO81
FFE8 60                         RTS
                        ;
                        ;       SOUND0-SOUND3 -- SET UP SOUND REGISTERS
                        ;
FFE9                    SOUND0
FFE9 A9 36                      LDA     #DROP           ;AUDF0
FFEB                    SOUND1
FFEB A0 04                      LDY     #PURE2          ;AUDC0
FFED                    SOUND2
FFED A2 0F                      LDX     #$0F            ;AUDV0 -- MAX VOL
FFEF                    SOUND3
FFEF 85 17                      STA     AUDF0
FFF1 86 19                      STX     AUDV0
FFF3 84 15                      STY     AUDC0
FFF5 60                         RTS

FFF6 00 00                      .BYTE   0,0             ;EXTRA BYTES

FFF8 00 00 00                   .BYTE   0,0,0,0         ;MORE EXTRA BYTES
FFFB 00

                        *=ROMSTR+$FFC
FFFC 00 F1                      .WORD   PSTART          ;START VECTOR
FFFE 00 00                      .BYTE   0,0             ;EXTRA BYTES
```

```
                ;
                ;       THE FOLLOWING CODE IS FOR DEBUGGING ONLY.
                ;       IT IS NOT TO BE INCLUDED IN THE FINAL CARTRIDGE.
                ;       THESE SUBROUTINES USE THE KIM MONITOR ROUTINES TO PRINT VARIOUS
                ;       INFORMATION ON THE TI TERMINAL CONNECTED TO THE DEVELOPMENT SYSTEM.
                ;
                        .IF     PRNT
                        *=AD

                        *=*-1/256+1*256
                ;
                ;       PRMOVE -- PRINT FROMB-TOSQRB
                ;
                PRMOVE
                        LDX     PRFLG           ;PRINT FLAG SET?
                        BEQ     PRRTN           ;NO.  DON'T PRINT
                        DEC     PRCNT           ;PRINT 10 MOVES TO THE LINE
                        BPL     PRMV10
                        JSR     CRLF
                        LDA     #10-1
                        STA     PRCNT
                PRMV10
                        LDX     FROMB           ;YES.
                        JSR     IEDSP           ;PRINT FROM SQUARE #
                        JSR     PRTBYT
                ;
                ;       PRTO -- PRINT TO SQUARE
                ;
                PRTO
                        LDA     PRFLG           ;PRINT FLAG SET?
                        BEQ     PRRTN           ;NO.  DON'T PRINT
                        LDA     #'-             ;YES.  PRINT '-'
                        JSR     OUTCH
                        LDX     TOSQRB          ;PRINT TO
                        TXA
                        BMI     PROT1           ;SKIP IF TO = BLANK (AA) OR JP (BB)
                        JSR     IEDSP           ;CONVERT FROM INTERNAL TO EXTERNAL NUMBERING
                PROT1
                        JSR     PRTBYT
                        JMP     OUTSP

                PRRTN
                        RTS
                ;
                ;       PRBRD -- PRINT ENTIRE BOARD & OTHER INFO
                ;
                PRBRD
                        LDA     PRFLG           ;PRINT FLAG SET?
                        BEQ     PRRTN           ;NO.  DON'T PRINT
                        LDA     #0              ;YES.  TURN OFF SOUND, IF ANY
                        STA     AUDV0
                        JSR     CRLF            ;YES.  PRINT CR

                        LDA     #GAMMSG         ;PRINT "GAME NUMBER ="
                        JSR     OUTAL3
                        LDA     GAMNO
                        JSR     PRTBYT
                        JSR     CRLF

                        LDA     #BLKMSG         ;"COMPUTER IS DARK"
```

VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM

```
                        LDA     #BLKMSG         ;"COMPUTER IS DARK"
```



```
                        LDX     COLCMP
                        CPX     #DARK
                        BEQ     AXB
                        LDA     #WHMSG          ;"COMPUTER IS WHITE"
                AXB
                        JSR     OUTMSG          ;PRINT MSG WITH CR
                        LDX     #B112           ;"DARK STARTS ON 1-12" (IF COLOR)
                        LDA     OLDPBQ
                        AND     #8
                        BNE     AXC             ;COLOR
                        LDX     #W112           ;B/W "WHITE STARTS ON 1-12"
                AXC
                        TXA
                        JSR     OUTMSG          ;PRINT MSG WITH CR
                        LDA     #TOPLFT         ;"TOP LEFT SQUARE IS"

                        JSR     OUTMS2          ;PRINT MSG WITHOUT CR
                        LDX     #34
                        JSR     IEDSP           ;COMPUTE TOP LEFT SQUARE # IN EXTERNAL FORM
                        JSR     PRTBYT

                        JSR     CRLF            ;PRINT BOARD
                        LDX     #34
                        JSR     PR2ROW
                        JSR     PR2ROW
                        JSR     PR2ROW
                        JSR     PR2ROW
                        JSR     PR2ROW
                PRBD70
                        JMP     CRLF

                ;
                ;       PR2ROW -- PRINT 2 ROWS AND DECREMENT X
                ;
                PR2ROW
                        JSR     BL2             ;2 BLANK SPACES
                        JSR     PRROW
                        JSR     PRROW
                        DEX
                        RTS
                ;
                ;       BL2 -- PRINT TWO SPACES
                ;
                BL2
                        TXA
                        PHA
                        JSR     OUTSP
                        JSR     OUTSP
                        PLA
                        TAX
                        RTS
                ;
                ;       PRROW -- PRINT ONE ROW (4 PIECES) & CR
                ;
                PRROW   JSR     PRPIEC
                        JSR     BL2
                        JSR     PRPIEC
                        JSR     BL2
                        JSR     PRPIEC
                        JSR     BL2
                        JSR     PRPIEC
```

```
                        TXA
                        PHA
                        JSR     CRLF
                        PLA
                        TAX
                        RTS
                ;
                ;       PRPIEC -- PRINT ONE PIECE (OR -- IF EMPTY)
                ;
                PRPIEC  TXA
                        PHA
                        LDA     BOARD,X
                        BNE     PRP10
                        LDA     #'-             ;EMPTY
                        JSR     OUTCH
                        LDA     #'-
                        JSR     OUTCH
                        JMP     PRP40
                PRP10
                        PHA                     ;NOT EMPTY
                        LDX     #'
                        TAY
                        BMI     PRP20           ;COMPUTER -- PRINT SPACE
                        LDX     #'H             ;HUMAN -- PRINT "H"
                PRP20
                        TXA
                        JSR     OUTCH
                        LDX     #'1
                        PLA
                        AND     #$7F
                        CMP     #KING           ;CHECKER = 1
                        BNE     PRP30           ;KING = 2
                        INX
                PRP30
                        TXA
                        JSR     OUTCH
                PRP40
                        PLA
                        TAX
                        DEX
                        RTS
                ;
                ;       OUTMS2 -- PRINT STRING AT MESSAG/256*256+A ON TI WITH NO CR
                ;
                OUTMS2
                        LDY     PRFLG           ;PRINT FLAG SET?
                        BEQ     PRRTN2          ;NO.  DON'T PRINT
                        TAX
                        LDA     TO              ;SAVE TO,TO+1
                        PHA
                        LDA     TO+1
                        PHA
                        STX     TO              ;MESSAGE ADR LSB
                        LDA     #MESSAG/256     ;MESSAGE ADR MSB
                        STA     TO+1
                        JSR     OUTM3
                        TAX
                        PLA                     ;RESTORE TO,TO+1
                        STA     TO+1
                        PLA
```

```
                        PLA
                        STA     TO
                        TXA
PRRTN2
                        RTS
;
;       OUTMSG -- SAME AS OUTMS2 EXCEPT DO CR
;
OUTMSG
                        LDY     PRFLG
                        BEQ     PRRTN2
                        JSR     OUTMS2
                        JMP     CRLF
;
;       PRMB -- INPUT:  Y=LSB OF ADDR OF MESSAGE TO BE PRINTED
;                       X=ADDR OF RAM PAGE ZERO VAR TO BE PRINTED
;               OUTPUT: X RESTORED, A,Y MODIFIED
;
PRMB
                        TXA
                        PHA
                        TYA
                        JSR     OUTMS2
                        PLA
                        PHA
                        JSR     PRTBYT
                        JSR     OUTSP
                        PLA
                        TAX
                        RTS
;
;       BLECCH -- CALL OUTALP, THEN PRALL
;
BLECCH
                        LDY     DEPTH
                        CPY     PRDEP
                        BCS     PRRTN2
                        JSR     OUTALP
;
;       PRALL -- PRINT A BUNCH OF JUNK FOR ALPHA-BETA PRUNING DEBUG
;
PRALL
                        JSR     PRMOVE
                        LDX     DEPTH
                        LDY     #QDEPTH
                        JSR     PRMB
                        LDX     ACTIVE
                        LDY     #QACTIVE
                        JSR     PRMB
                        JSR     OUTSP

                        LDX     DEPTH
                        LDA     ALPHAH-1,X
                        JSR     PRTBYT
                        LDX     DEPTH
                        LDA     ALPHAL-1,X
                        JSR     PRTBYT
                        JSR     OUTSP

                        LDX     DEPTH
                        LDA     ALPHAH,X
```

```
                        JSR     PRTBYT
                        LDX     DEPTH
                        LDA     ALPHAL,X
                        JSR     PRTBYT
                        JSR     OUTSP

                        LDX     DEPTH
                        LDA     ALPHAH+1,X
                        JSR     PRTBYT
                        LDX     DEPTH
                        LDA     ALPHAL+1,X
                        JSR     PRTBYT
                        JSR     OUTSP

                        JSR     CRLF
                        LDA     PRDEP
                        CMP     #4
                        BCC     PRA20
                        LDX     DEPTH
                        CPX     #2
                        BEQ     PRA10
                        CPX     #1
                        BNE     PRA20
                        JSR     CRLF            ;2 SPACES IF LEVEL 1
PRA10
                        JSR     CRLF            ;1 SPACE IF LEVEL 2
PRA20
                        RTS
;
;       OUTAL3 -- OUTPUT MSG A WITHOUT CR
;
OUTAL3
                        LDY     #MJ/256
                        STA     TO
                        STY     TO+1
;
;       OUTM3 -- MESSAGE PRINTING LOOP
;
OUTM3
                        LDY     #0
                        TYA
                        PHA
OUTM10
                        LDA     (TO),Y
                        BEQ     OUTM20          ;END OF MESSAGE IS INDICATED BY A 0 BYTE
                        JSR     OUTCH
                        PLA
                        TAY
                        INY
                        TYA
                        PHA
                        BNE     OUTM10          ;JMP
OUTM20
                        PLA
                        RTS
;
;       OUTALP -- OUTPUT MESSAGE A WITH CR
;
OUTALP
                        JSR     OUTAL3
```

```
                        JMP     CRLF
                ;
                ;       PASK -- INPUT NEW PRFLG AND PRDEP FROM TI
                ;
                PASK
                        LDA     #1
                        STA     PRFLG
                        JSR     CRLF
                        LDA     #PRMSG
                        JSR     OUTMS2
                        JSR     GETBYT
                        PHA
                        JSR     CRLF
                        LDA     #PRDMSG
                        JSR     OUTMS2
                        JSR     GETBYT
                        STA     PRDEP
                        PLA
                        STA     PRFLG
                        JMP     CRLF
                ;
                ;       PRMOV2 -- SIMILAR TO PRMOVE
                ;
                PRMOV2
                        LDA     CURSOR
                        STA     FROMB
                        LDA     PRFLG
                        AND     #$FE            ;DO CR'S IF PRINTING ALPHA-BETA JUNK
                        BEQ     JOY800
                        JSR     CRLF
                        JSR     CRLF
                        JSR     PRMOVE
                        JSR     CRLF
                        JSR     CRLF
                        LDX     MOVE
                        RTS
                JOY800
                        JSR     PRMOVE
                        LDX     MOVE
                        RTS
                        .PAGE

                        *=*-1/256+1*256         ;GO ON TO NEXT PAGE BOUNDARY
                MESSAG
                PRMSG   .BYTE   "ENTER PRFLG?",0
                PRDMSG  .BYTE   "ENTER PRDEP?",0
                GAMOVR  .BYTE   "GAME OVER",0
                BLKMSG  .BYTE   "COMPUTER IS DARK",0
                WHMSG   .BYTE   "COMPUTER IS WHITE",0
                B112    .BYTE   "DARK STARTS ON 1-12",0
                W112    .BYTE   "WHITE STARTS ON 1-12",0
                TOPLFT  .BYTE   "TOP LEFT SQUARE IS ",0
                        .PAGE
                JLERR   .BYTE   "JOYLP ERROR",0
                QDEPTH  .BYTE   "DEPTH=",0
                QINDEX  .BYTE   "INDEX=",0
                QALPHA  .BYTE   "ALPHA(DEPTH)=",0
                QBETA   .BYTE   "ALPHA(DEPTH-1)=",0
                QALP1   .BYTE   "ALPHA(DEPTH+1)=",0
                QHPIECE .BYTE   "HPIECE=",0
```

```
QCPIECE .BYTE   "CPIECE=",0
QACTIVE .BYTE   "ACTIVE=",0
        .PAGE

        *=*-1/256+1*256
MJ      .BYTE   "MULTIPLE JUMP ",0
MJBA    .BYTE   "FRSMOV & BETTER ALPHA",0
WA      .BYTE   "WORSE ALPHA",0
PRUNE   .BYTE   "PRUNE",0
BANFM   .BYTE   "BETTER ALPHA NOT FRSMOV",0
BAFM    .BYTE   "BETTER ALPHA FRSMOV",0
        .PAGE
GAMMSG  .BYTE   "GAME NUMBER=",0
RAN     .BYTE   "RANDOM ",0


        *=*-1/256+1*256
;
;       PRINTING VARS FOR DEBUG
;
PRCNT
        *=*+1           ;COUNT OF NUMBER OF MOVES PRINTED ON A LINE
PRFLG
        *=*+1           ;PRINTING FLAG
PRDEP
        *=*+1           ;MAX DEPTH TO BE PRINTED+1
        .ENDIF
        .END
```

VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM

SYMBOL TABLE

AB10	F4F6	AB20	F4FE	AB30	F500	AB50	F2CD
ABD2	F50A	ABDONE	F515	ABRET1	F51E	ABRET2	F3C4
ACTIVE	00BB	ADD10	FE2B	ADDSUM	FE1A	ADDTAB	FFC8
ADDTRM	0020	ADS10	FD19	AJ05	F333	AJ10	FE57
AJJ5	FE69	AJUMP	F328	ALJ	F4E1	ALOOP1	F2D7
ALP200	FBE5	ALP205	FBFC	ALP210	FC0B	ALP220	FC0D
ALP230	FC11	ALP231	FC12	ALP235	FC13	ALP240	FC1D
ALPEND	00F8	ALPH05	F2E2	ALPH10	F2E7	ALPHAH	00EB
ALPHAL	00DE	ALPHBE	F2BC	ALPJMP	F376	ANYMVE	00B6
ARTN	FA8C	ATIM	00EE	ATTR10	F7D6	ATTR20	F7E1
ATTR30	F7EF	ATTR35	F7F4	AUDC0	0015	AUDC1	0016
AUDF0	0017	AUDF1	0018	AUDV0	0019	AUDV1	001A
BACK10	FD45	BACK20	FD79	BACK25	FD16	BACK30	FD2A
BACK40	FD30	BACK50	FD36	BACK60	FD3C	BACK70	FD58
BACK80	FD60	BACK90	FD68	BACK95	FD70	BAKTAB	FF52
BLANK	00AA	BLKJMP	F1CD	BLNKTM	00F0	BNEQ	F6D7
BOARD	0080	BRDCHR	FF00	BRDLP	F994	BSWLP	F19F
CENT50	FDDD	CENT90	FDE3	CHECKE	0010	CHGCNT	FE51
CKING	00AF	CKTAB	FF45	CLP0	0000	CLP1	0080
CLRLP	F974	CMPCHK	0090	CMPCRS	00B0	CMPKNG	00A0
CMPMVE	F243	COL0	00D0	COL1	00D2	COL2	00D4
COL3	00D6	COLCMP	00B2	COLHUM	00B1	COLP0	00CE
COLP1	00CF	COLSQ	00F3	COLTB	FFCB	COLUBK	0009
COLUP0	0006	COLUP1	0007	COLUPF	0008	COMP05	F213
COMP06	F238	COMP07	F23C	COMP10	F240	COMPU2	F2BC
COMPUT	F28D	CONT1	F4D0	CONT2	F4D2	CONT2J	F330
CONT3	F4E4	CONT3J	F2DF	CORN	0004	CORN20	FD91
CORN30	FDA0	CORN40	FDA9	CORN50	FDB4	CORN60	FDBF
CORN70	FDC8	CPIECE	00AA	CRLF	1E2F	CRN100	FDD3
CRSB03	F78E	CRSB10	F793	CRSB13	F7A3	CRSB14	F7B3
CRSB15	F7BB	CRSB17	F7BD	CRSB20	F7D4	CRSBLK	F766
CRSTUP	FAFB	CTAB	FF40	CTLSWA	0281	CTLSWB	0283
CTRLPF	000A	CURS	0030	CURSC	00E8	CURSOR	00AB
CXBLPF	0036	CXCLR	002C	CXMOFB	0034	CXMOP	0030
CXM1FB	0035	CXM1P	0031	CXPOFB	0032	CXP1FB	0033
CXPPMM	0037	DARK	0036	DEN	00B8	DEPTH	00BC
DIAG	FF7E	DIAG10	FD0B	DIAG20	FD0D	DIAGLP	FCF8
DIS02	FC6C	DIS03	FC6F	DIS04	FC76	DIS05	FC8F
DIS30	FCBC	DIS40	FCD2	DIS50	FCD8	DIS60	FCDF
DIS65	FCE2	DIS67	FCF2	DIS70	FD10	DISLP1	FC78
DISLP2	FCA6	DOCENT	FDD3	DOROT	F842	DRKCHR	0092
DROP	0036	EDARK	0000	ELIAT	0036	ELIGHT	0034
EMPTY	0000	ENABL	001F	ENAM0	001D	ENAM1	001E
ENDPC	0008	ERRSND	000F	FH20	F4B5	FH21	F4CB
FH5	F465	FH6	F498	FH7	F49B	FHBLEC	F4BF
FHQ	F4B7	FINSH3	F44A	FRAM10	F14E	FRAME	00C3
FRAMLP	F125	FRAQ	F47C	FREEZE	00F2	FRM10	F71F
FRM15	F751	FRM20	F75B	FRM30	F762	FRMLP2	F140
FROM	00F4	FROM10	F63D	FROMB	00AC	FROMT	00C8
FRQMTO	00E7	FRSMOV	00BF	FRSTAB	FF6E	FRZTIM	002D
FSET20	FB07	FSET30	FB0D	GAMNO	00B0	GETBYT	1F9D
GETCH	1E5A	GFLG2	00EF	GRP0	001B	GRP1	001C
GRTN2	F96C	GS1	F96D	GSCON	002D	GSKIP	F968
GSRST	F944	GSTIM	00EB	HDARK	00B0	HIDIGT	F82E
HKING	00AE	HKTAB	FF43	HLIAT	0026	HLIGHT	0026
HMBL	0024	HMCLR	002B	HMIN	0005	HMM0	0022
HMM1	0023	HMOVE	002A	HMP0	0020	HMP1	0021
HPIECE	00A9	HTAB	FF3E	HUMCHK	0010	HUMCRS	0030
HUMKNG	0020	IED02	FA98	IED04	FA9A	IED05	FAAB

IED07	FAAF	IED10	FAB6	IED20	FAC5	IEDLP1	FABA
IEDLP2	FAC3	IEDSP	FA8D	ILEGAL	00ED	INDEX	00BE
INLP	F105	INPT0	0038	INPT1	0039	INPT2	003A
INPT3	003B	INPT4	003C	INPT5	003D	INTIM	0284
JMP02	FA51	JMP05	FA55	JMP10	FA6A	JMP12	FA79
JMPBAK	F000	JMPCHK	F9FE	JMPCK2	FA06	JMPEND	00CE
JMPF2	FA1F	JMPI	00A4	JMPLEN	0009	JMPLP1	FA10
JMPLP2	FA3E	JMPLST	00C5	JMPRTN	F9FD	JMPS10	F9DE
JMPS20	F9E2	JMPSAV	FE9F	JMPSET	F9CE	JMPSLP	F9D2
JMPSQR	00BD	JMPSR	F9E7	JMPSV2	FE6A	JMPWIN	F9E8
JOY10	F5EA	JOY11	F5FC	JOY111	F610	JOY12	F658
JOY13	F65A	JOY20	F667	JOY3	F5B3	JOY81	F6CA
JOY9	F5C5	JOY90	F6CE	JOYLP	F5BD	JOYTAB	FF62
JP	00BB	JPF1	FA19	JPM05	F613	JPM06	F61D
JPM10	F621	JTHRU	F54C	JTHRU2	F56E	JUMP1	FB80
JUMP4	FBB5	JUMPL	FBBB	JUMPR	FBCD	JUMPS	FBA5
K100	F09C	K110	F0A2	K130	F0B6	KING	0020
KNGCHK	FFDF	KNGCK2	FEBC	KRN10	F080	KRN15	F086
KRNLP	F072	LB	FFFC	LF	0005	LIGHT	000C
LINCHR	0010	LOADY	FA22	LODRAM	F809	LODY10	FA2C
LOOP1	F06E	LOOP15	F07E	LOOP2	F0BC	LOOP20	F0D0
LP06	F923	LP10	F925	LPQX	F8F7	LPSKIP	F8D6
LPX	F8E8	LTCHR	000C	M28LP	F6FC	MAXMAX	000B
MJ05	F36A	MJ07	F387	MJ08	F38C	MJ15	F39B
MJ20	F3A8	MLT	0008	MLTFLG	00A8	MLTJMP	00A6
MOVCHK	FA2D	MOVCLR	FEE1	MOVE	00E9	MOVE10	F6D4
MOVE12	F6DD	MOVE20	F6E6	MOVE28	F6EC	MOVE29	F6FA
MOVE36	F704	MOVE37	F707	MOVE40	F70C	MOVE50	F711
MOVE60	F731	MOVE70	F749	MOVFLG	00E5	MOVTAB	FF66
MOVVAL	00E4	MSKLST	FF4A	MST	0004	MSTFLG	00A7
MSTJMP	00A5	MV285	FAEA	MV287	FAEC	MVTAB2	FF6A
NEW10	F456	NEW40	F491	NEW50	F43B	NK	FB96
NOBL	F1F2	NOCHCK	F5A7	NOCHDO	F624	NOJUMP	F55A
NOK	FB46	NOK10	FB5A	NOR05	F190	NOR06	F19B
NOREV	F1CA	NORLP	F188	NOTON	FAE0	NUM	00B5
NUSIZ0	0004	NUSIZ1	0005	NXTAB	FF4E	OLDINP	00EA
OLDPB6	00EC	OLDPBQ	00C2	ONBRD2	FAC9	OTHER	00D3
OUTCH	1EA0	OUTSP	1E9E	OVRCON	0025	OVSLP	F1CF
PAL	0000	PF0	000D	PF1	000E	PF2	000F
PICKUP	0030	PIECE	00A3	PLAY2	0010	PNTR0	00D8
PNTR1	00DA	PNTR2	00DC	PNTR3	00DE	PRNCNT	00C4
PRNJMP	F419	PRNT	0000	PRTBYT	1E3B	PSTART	F100
PURE2	0004	QMV10	FE0E	QMV100	FE1A	QMVLP	FE05
QQ10	F885	QQ20	F8B2	QQ30	F8B9	QQ40	F8C8
QQ50	F90F	RANTAB	FEEA	RB	FFFB	REFP0	000B
REFP1	000C	RES10	F9A5	RES15	F9B7	RES20	F9C9
RESBL	0014	RESET	F982	RESET2	F95E	RESLP	F97D
RESM0	0012	RESM1	0013	RESMP0	0028	RESMP1	0029
RESP0	0010	RESP1	0011	RESTOR	FB62	RESTR4	FB1F
RESTR5	FB3C	RESTR6	FB1B	RF	0004	ROMSTR	F000
RDTIT	F846	RSV	F58A	RSV2	F58D	RSYNC	0003
SAVE1	FEAF	SCOR2	F002	SCOR3	F004	SCOR4	F019
SCP0	00F6	SCP1	00F7	SCRLP0	00D0	SCRLPF	00DA
SCRRP1	00D5	SCRRPF	00DF	SCRTBL	FF8C	SED05	FC54
SED07	FC5F	SED20	FE3A	SED25	FE42	SED30	FE4F
SEDONE	FC3C	SELP	FC2C	SKIP2	F444	SNDTIM	00F1
SOUND0	FFE9	SOUND1	FFEB	SOUND2	FFED	SOUND3	FFEF
SQREND	00E1	SQUARE	00E0	STACK	0000	STATEV	FC1D
STUP10	F24A	STUP25	F25F	STUP30	F263	STUP35	F270
STUP40	F287	STUP50	F28A	SW10	F175	SWCHA	0280

VCS CHECKERS BY CAROL SHAW 5/15/80 9:41 AM

SWCHB	0282	SWIT	FB14	SWLP	F16E	SYSTEM	FF75
T0	00B3	T1	00B4	T1024T	0297	T2	00B5
T3	00B6	T4	00CE	T5	00CF	T6	00B7
T7	00B9	T8	00BA	TER	F31B	TERMNO	FBCE
TERR	FB73	TIM1T	0294	TIM64T	0296	TIM8T	0295
TIMOUT	00C1	TO10	F673	TO11	F68B	TO50	F6B8
TO51	F6BB	TO60	FECA	TO70	FED6	TO80	FEDC
TO81	FFE8	TOMOV	FEDE	TONORM	FAF3	TOSET	FAE2
TOSQR	00F5	TOSQRB	00AD	TOSUB	FFD1	VBLANK	0001
VBLCON	002C	VBLP	F938	VBTIM	001E	VDELBL	0027
VDELP0	0025	VDELP1	0026	VSYNC	0000	WASJ	F40D
WASJ10	F421	WASJ17	F441	WASJ18	FE7B	WASJ90	FE93
WASJLP	FE8A	WIN60	F20E	WIN70	F210	WINSAV	00E6
WSLP	F865	WSYNC	0002	XIE1	FA7D	XMOVE	00C0
XSAVE	00B4	YSAVE	00B5	ZROBGN	0080	ZROEND	00B0